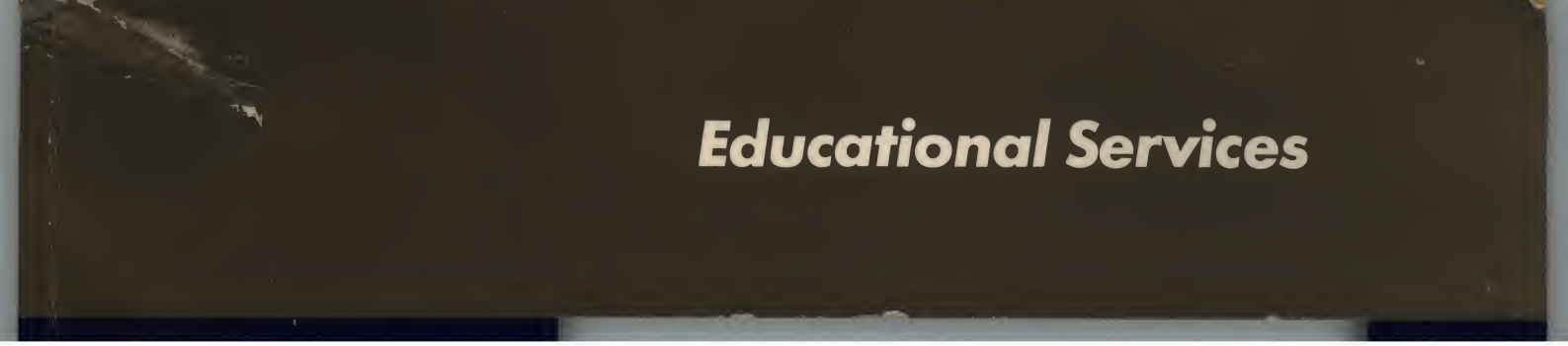

Educational Services

digital

EY-L055E-HO H001

MicroVMS
OPERATIONS AND MANAGEMENT

STUDENT GUIDE

Tables

INTRODUCTION TO THE MicroVAX II

Figures

Tables

SYSTEM START-UP AND PRODUCT INSTALLATION

Figures

Tables

Examples

USING REMOVABLE STORAGE MEDIA

Figures

Tables

Examples

SYSTEM MANAGEMENT FUNCTIONS

Tables

Examples

SETTING UP AND USING A NETWORK

Figures

Examples

Student Guide

Introduction

MicroVMS Operations and Management is a course designed to teach operational and management functions for a MicroVAX II system running MicroVMS Version 4.2 or later.

The Student Workbook is divided into a number of units, or modules, each designed to cover a well-organized topic, or group of topics. Most modules are divided into smaller units, each with its own examples.

This **Student Guide** module describes the contents of the course, and suggests ways in which you can most effectively use its materials. This guide discusses the following topics:

- Course Description
- Course Prerequisites
- Course Goals and Nongoals
- Course Resources
- Course Organization
- Course Map
- Course Conventions

1 Course Description

This course is intended for MicroVAX users who need to learn operational and management functions of a MicroVAX II system running MicroVMS. The course presents the management and operational functions that enable you to manage a single MicroVAX with a single user, or several MicroVAXes with several users.

Among the topics addressed are:

- The MicroVAX/MicroVMS hardware and software environment
- System start-up and shutdown, and product installation
- Removable storage devices, including the RX50 diskette drive and the TK50 tape drive
- System management functions, such as:
  - Adding and deleting user accounts
  - Managing batch and print queues
  - Backing up user files
- Managing a network

2 Prerequisites

To derive the full benefit from this course, you should be able to:

- Log in and log out on a MicroVAX/MicroVMS system
- Perform file manipulation tasks, such as copying, renaming, and deleting files
- Manage files in a directory structure
- Use a text editor to create and edit text files
- Use VMS utilities such as Mail
- Write simple command procedures such as LOGIN.COM

You can obtain these skills by taking the lecture/lab or self-paced *VAX/VMS Utilities and Commands* course.

3 Course Goals

This course is intended to prepare you to:

- Identify the various hardware peripherals and layered products associated with the MicroVAX II system
- Start up and shut down the MicroVAX II system
- Install MicroVMS and other layered products on the system, using RX50 or TK50 distribution media
- Perform system management functions necessary to maintain and monitor the system
- Configure the system into a DECnet network

4 Course Nongoals

This course does not address the following topics:

- Basic user-level information, such as manipulating files or writing simple command procedures
- System tuning and other system performance issues that require extensive use of the SYSGEN and AUTOGEN utilities
- Program development issues, including the use of utilities such as the VAX/VMS Symbolic Debugger, MACRO assembler, MONITOR, and the error log report generator

5 Equipment/Special Tools

You should have access to a MicroVAX II system and an installation kit for the most recent version of the MicroVMS software.

6 Course Resources

In addition to the MicroVAX system itself, there are several major resources that should be available to you as you complete this course:

1. This **Student Workbook**

You should also have copies of the following manuals while taking the course:

- *MicroVMS User's Manual*
- *MicroVAX II Owner's Manual*

In addition, you may find these other manuals useful:

- *Guide to Networking on VAX/VMS*
- *Guide to VAX/VMS System Security*
- *Introduction to Local Area Networks*

7 Course Organization

A module contains the following instructional elements:

- An **introduction**, which describes the purpose of the unit, provides some motivation for mastering its objectives, and outlines its contents.
- One or more **objectives**, which describe the operations for which the module provides instruction. Objectives are designed to focus your study efforts on a selected number of skills.
- A list of **resources** required to complete the unit. Some of these resources are distributed with this course; others are not.
- A package of **instructional materials** to introduce the skills described by the objectives, including:

  Descriptive text – The text of a module or section ties various elements together. Text describes the importance of a particular set of elements or skills, lists the occasions when you are most likely to use them, or breaks a complex operation into a number of distinct steps.

  Figures – Figures illustrate hardware elements, concepts, or logical relationships presented in the module.

  Tables – Some tables summarize the operations introduced in a particular section; others help you to distinguish between related concepts.

  Examples – The examples illustrate how to use the concepts and operations a lesson describes.

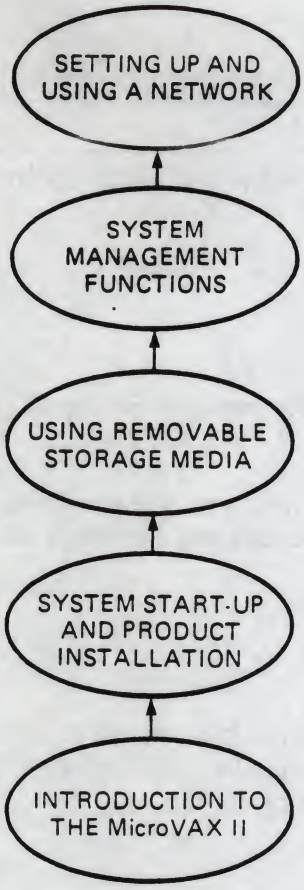
- A **summary** of the topics presented in the unit.

Begin a new module by reading its introduction and objectives. Work through the module at your own pace, completing readings and exercises in the recommended order.

The examples and exercises are powerful instructional elements in the course that provide you with actual operations and concepts. They also give you ample opportunity to practice.

Regardless of your level of experience, you are more likely to master the objectives of a unit by reading examples and performing exercises in addition to reading documents. Plan to spend as little time as possible at your desk, and as much time as possible at a terminal.

8 Course Map



MKV85-1581

9 Course Conventions

This table describes the conventions used in the listings and command tables of the Student Workbook.

Table 1 Course Conventions

Convention	Meaning

Examples	Examples of user input and system output appear in a different typeface.
	
	For example:
	
	`$ SHOW TIME`
	`05-MAR-1984 11:55:22`

CTRL/X	Press and hold the key labeled CTRL while you press another key (X). Many control keys have special meanings.

UPPERCASE	In commands, uppercase characters indicate words you type exactly as they appear. For example, you would type the following commands as they appear:
	
	`$ DIRECTORY`
	`$ TYPE LOGIN.COM`

lowercase	Lowercase characters represent elements that you must replace according to the description in the text. For example, you must follow certain rules when you replace "file-spec" in the following example: `$ TYPE file-spec`
Ellipsis (. . .)	Horizontal ellipses indicate that you can enter additional parameters, values, or information. For example, you can enter any number of file specifications in the following example: `$ TYPE file-spec,...`
Square brackets ([])	Square brackets indicate that the enclosed item is optional. (Square brackets are not optional, however, in the syntax of some file specifications.) For example, the logical name is optional in the following command: `$ MOUNT/FOREIGN $TAPE1 [logical-name]`

INTRODUCTION TO THE MicroVAX II

Introduction

The MicroVAX II is a member of the VAX family of computers. Members of the VAX family vary in power and size, from microcomputers to large super-minicomputers and VAXcluster systems. All the systems in the VAX family are compatible with each other; they can run the same software, and they can share data with each other by means of a network. Since the advent of the first VAX system, the VAX-11/780, DIGITAL has developed a wealth of software for the VAX family.

The MicroVAX II processor brings the power of a VAX to the microcomputer user. It provides almost the same computational power as the VAX-11/780, and is a fraction of the size and cost of the VAX-11/780. The same operating systems available on the larger VAX systems are supported on the MicroVAX II, as are most VAX/VMS optional products.

Your MicroVAX II system consists of two basic components: hardware and software. To effectively operate and manage your system, you should know the functions and capabilities of each major hardware and software component.

This module introduces the hardware and software of the MicroVAX/MicroVMS system. It discusses some components of the MicroVAX II hardware, including:

- The MicroVAX cabinets, and their controls and indicators
- The Central Processing Unit (CPU)
- Main memory
- Input/Output (I/O) controllers and peripheral devices

It also discusses some of the software that is available for the MicroVAX II, including:

- The MicroVMS operating system
- Optional software, which may be DIGITAL products or user-written programs

Objectives

To effectively operate and manage a MicroVAX II system with MicroVMS, you should be able to:

- Identify the functions of the important parts of the MicroVAX hardware, including:
  - The processor control panel, I/O connection panel, and CPU distribution panel on the MicroVAX cabinet
  - The indicators and controls on the MicroVAX processor control panel and CPU distribution panel
  - The Central Processing Unit (CPU)
  - Main memory
  - The Q-Bus
  - Peripheral devices, such as terminals, printers, and mass storage
- State some of the features of the MicroVAX architecture
- State some of the features and capabilities of the MicroVMS operating system, including:
  - Some of the utilities available with MicroVMS
  - The modular packaging of the operating system
  - Some of the optional products supported by MicroVMS

Resources

The following documents contain information on the hardware and software of the MicroVAX/MicroVMS system:

1. *MicroVAX II Owner's Manual*
2. *VMS System Software Handbook*

1 MicroVAX II Hardware

Every computer system has certain essential components. They include a Central Processing Unit (CPU) for executing programs, main memory for temporary storage of programs and data, peripheral devices including terminals and mass storage devices, and a path by which these components communicate. The MicroVAX II has all of these components, except for some peripheral devices, located inside its cabinet.

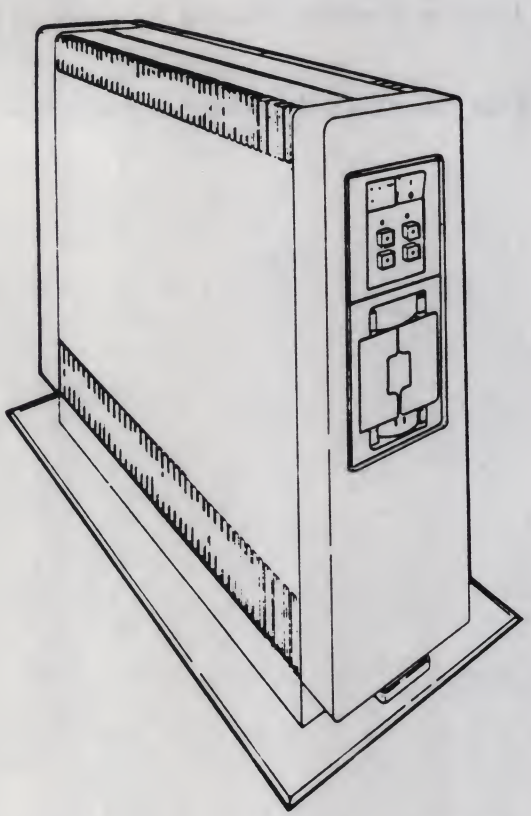
On the outside of the MicroVAX II cabinet are control switches, indicator lights, and a place where external peripheral devices can be connected to the MicroVAX II. Several cabinets are available for the MicroVAX II, and the locations of external features vary with the type of enclosure.

The following sections discuss the parts of the MicroVAX II hardware in more detail.

1.1 MicroVAX II Enclosures

The MicroVAX II system is available in these enclosures:

- The BA23 (pedestal or table-top) enclosure, shown in Figure 1. The BA23 is also available as a rack mount without the outer shell.
- The BA123 (floor-standing) enclosure, shown in Figure 2. The BA123 has more space for installing peripheral devices, and more power capacity, than the BA23.



MKV85-1549

Figure 1 MicroVAX II in BA23 Enclosure

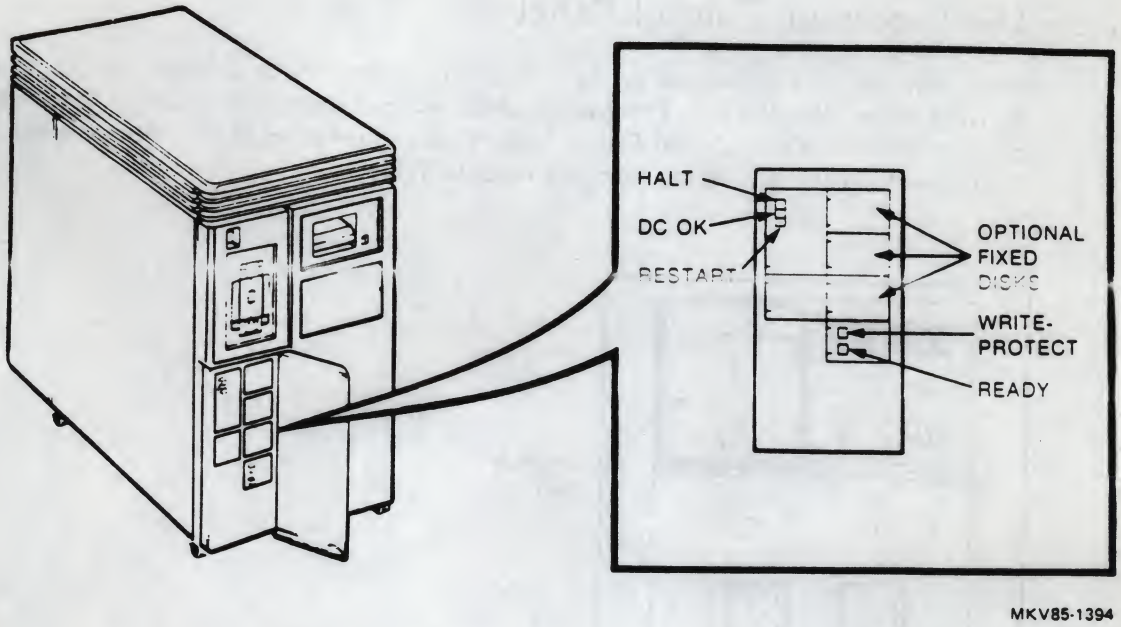


Figure 2 MicroVAX II in BA123 Enclosure with Control Panel

1.2 The Processor Control Panel

The system control switches and indicator lights are located on the processor control panel on the front of the MicroVAX II. The location of the processor control panel on the floor-standing model is shown in Figure 2, and Figure 3 shows the control panel of the pedestal model. The controls and indicators have the functions listed in Table 1.

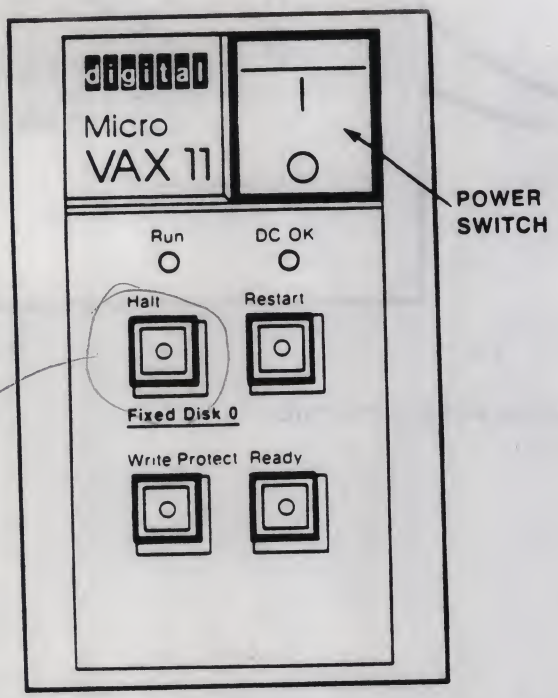


Figure 3 The BA23 MicroVAX II Processor Control Panel

Table 1 Control Switches and Indicator Lights on a MicroVAX II System

Switch or Indicator	Function
System Power Switch	Turns ac power on and off. On the pedestal model, the power switch glows orange when the power is on.
Halt Pushbutton	Stops the system and puts the CPU in console mode. Glows red when pressed in. Press again to release.
Restart Pushbutton	Attempts to restart the system when it is halted.
Write-Protect Pushbutton	When pressed in, prevents data from being written to the fixed disk. Glows orange when pressed in.
Ready Pushbutton	Prevents the fixed disk from being read from or written to when pressed in. Glows green when the disk is available (button is out); no color when button is pressed in.
Run Indicator	Glows green when the system is running.
Dc OK Indicator	Glows green when the dc voltages are correct.

Module 2 gives examples of when to use the Halt and Restart pushbuttons.

While the system is running, never press in the Write-Protect or Ready pushbutton for your system disk. These actions would prevent the operating system from writing information on the disk, possibly leaving system files in an inconsistent state.

1.3 The I/O Connection Panel and CPU Distribution Panel

The I/O connection panel is on the rear of the cabinet. It has removable inserts that can be replaced with connectors for peripheral devices. You plug external devices, such as additional terminals, into connectors on the I/O connection panel.

The CPU distribution panel takes up one insert of the I/O connection panel. It has controls that affect the operation of the system.

Figure 4 shows the rear of the pedestal cabinet, with the cover removed. The *MicroVAX II Owner's Manual* for your configuration tells you how to open the rear cover for access to the I/O connection panel. The I/O connection panel takes up most of the back of the cabinet. The enlarged portion at the right of the figure shows the CPU distribution panel. The controls on the CPU distribution panel have the functions listed in Table 2.

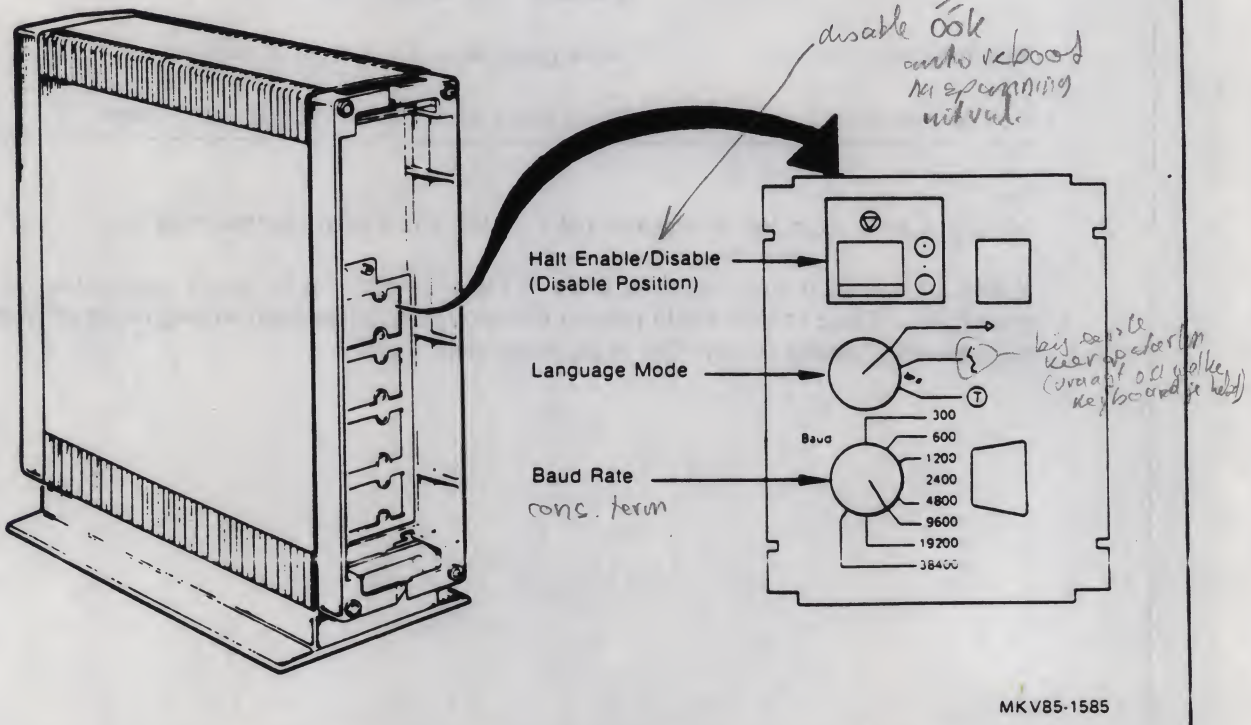


Figure 4 Rear of BA23 Cabinet, Showing I/O Connection Panel and CPU Distribution Panel

Table 2 Controls on the CPU Distribution Panel

Control	Function
Halt Enable/Disable	When in the "disable" position, disables the HALT button on the control panel and allows the system to restart automatically.
Language Mode	Selects the language to use when the CPU is in console mode. Place in the "inquire" (middle) position before you start the system for the first time. After choosing a language, leave in the arrow (top) position. (Module 2 explains the use of this switch.)
Baud Rate	Controls the speed at which characters are transmitted to and from the console terminal. Should be set to 9600. The baud rate set on the terminal must match the setting of this switch. See the owner's manual for your console terminal to find out how to set its baud rate.

1.4 The Central Processing Unit (CPU)

The MicroVAX II CPU module is a single board inside the MicroVAX II cabinet. This board contains:

- The CPU itself on a single chip.
- A chip for performing arithmetic on floating-point numbers.
- Either 256 kilobytes or one megabyte of main memory (a kilobyte is equivalent to about 1000 characters of text, and a megabyte is equivalent to about a million characters).
- Read-Only Memory (ROM) containing the bootstrap program. This program starts executing when you turn the MicroVAX on. It performs some tests on the MicroVAX hardware, then reads the MicroVMS operating system into memory from the disk (either automatically or in response to a console command).
- A time-of-year clock, for keeping track of the date and time. If you turn off power to the system, a battery keeps the clock running for up to ten days.

At any time, the CPU can be in either of two modes:

1. Program mode, in which it executes the operating system and user programs. This is the normal mode.
2. Console mode, in which it responds to special commands; for instance, commands to start up or halt the system.

When the CPU is in console mode, you must use the console terminal to enter commands. This terminal is plugged into the connector on the CPU distribution panel. Besides using it to enter console commands, you can use the console terminal as an ordinary user terminal. Later, you will learn how to switch the CPU between program and console mode, and to use some of the console commands.

1.5 Main Memory

In addition to the memory on the CPU board, you can have up to two additional memory modules installed in your MicroVAX II. There are three types of modules: one-megabyte, two-megabyte, and four-megabyte modules. Thus, the maximum amount of main memory the MicroVAX II can support is nine megabytes.

Some computers need a large amount of main memory to run large programs. This is not the case with the MicroVAX II running MicroVMS, because the MicroVMS operating system implements **virtual memory**: almost any program can run on a system with any amount of memory. However, additional memory allows your system to perform more efficiently or support more users.

1.6 The Q-Bus

The **Q-Bus** is the MicroVAX II's internal communication path. The CPU and the peripheral devices communicate with each other via the Q-Bus. Components are attached to the Q-Bus by being plugged into slots inside the MicroVAX II. The CPU module is plugged into a slot, as is each I/O controller module (see Section 1.7).

The MicroVAX II, unlike some other systems that use the Q-Bus, does not use the Q-Bus for communication between the CPU and main memory. If there are separate memory modules, they are connected directly to the CPU by a special cable.

1.7 I/O Hardware

On the MicroVAX II, some peripheral devices are located inside the cabinet and some outside. In either case, the device is usually connected by a cable to a separate controller module. The controller module is plugged into a slot inside the cabinet.

Section 1.8 summarizes the peripheral devices currently available for the MicroVAX II.

1.8 MicroVAX II Peripheral Devices

Terminals let users enter commands to the MicroVAX II and receive responses from the system. The MicroVAX II supports many terminals including:

- VT241 – color graphics and text
- VT240 – black-and-white graphics and text
- VT220 – text only
- VT100 – text only
- LA120, LA210, LA50 – hard-copy graphics and text
- LQP03 – hard-copy text

On a single-user system, you need only the console terminal. To support more users, you connect more terminals to the MicroVAX II with controller modules, such as:

- DZQ11 – supports four terminals
- DHV11 – supports eight terminals

Disk drives are used for storage of large amounts of software or data. The MicroVAX II has access to at least one disk drive. Drives that can fit inside the MicroVAX II cabinet include:

- RD51 – 10 megabytes
- RD52 – 31 megabytes
- RD53 – 71 megabytes

The MicroVAX II also needs a removable mass storage device. Removable media are used when installing the operating system and other software, for making backup copies of data, and for transferring data from one system to another. Devices that use removable media include:

- TK50 – drive for 95-megabyte tape cartridges
- RX50 – dual drive for 400-kilobyte floppy diskettes

The MicroVAX II has at least one of these devices inside its cabinet, and can have both.

Two other optional storage devices, each having its own table-top enclosure, are:

- RC25 - drive for one 26-megabyte fixed disk and one 26-megabyte removable disk
- RRD50 (CD Reader) - drive for 600-megabyte compact disks (compact disks are read-only)

To connect the MicroVAX II to a DECnet network, you need additional hardware such as:

- DEQNA - connects the MicroVAX to an Ethernet local area network
- DMV11 - connects MicroVAX systems directly to each other, or to large VAX systems, locally or long-distance
- Terminal lines can also be used for DECnet

Module 5 discusses network hardware in more detail.

The VAXstation II is a MicroVAX II with added workstation hardware:

- QVSS - video subsystem, which includes a keyboard, a large-screen graphics display, and a pointing device called a mouse

The VAXstation II also comes with additional workstation software that runs under MicroVMS. Managing a VAXstation II is almost the same as managing a MicroVAX II, thus everything you learn in this course applies to the VAXstation II.

MicroVMS will support many new peripheral devices as they become available. The amount of additional hardware you can add to the MicroVAX II depends on how many slots are available for hardware options and how much power each option uses. For an up-to-date list of MicroVAX II pre-configured systems and supported peripherals, and information on configuring a MicroVAX II system:

- Refer to the latest *VAX Systems and Options Catalog* (issued quarterly)
- Talk to your DIGITAL sales representative

2 The MicroVAX Architecture

The **architecture** of a computer system is the defined behavior of its hardware. The VAX architecture defines the behavior of the processors in the VAX family. The architecture of a processor is independent of the type of hardware used to build the processor. VAX processors are built using different hardware technologies, but they can all run the same software because they all implement the VAX family architecture.

The MicroVAX II supports a subset of the VAX architecture. In other words, the MicroVAX II hardware performs most but not all of the functions defined by the VAX architecture.

One aspect of the architecture of a computer is the set of instructions the CPU can execute. The MicroVAX II processor can execute most of the instructions defined by the VAX architecture. When a program contains an instruction that the processor cannot perform, the MicroVMS operating system software handles the instruction instead. This action is called **emulation**. The instructions emulated by the operating system include some operations on character strings and floating-point numbers.

The VAX architecture defines a set of PDP-11 compatibility mode instructions, for executing software written for the DIGITAL PDP-11 computers. The MicroVAX II does not execute these instructions, and MicroVMS does not emulate them. However, the VAX RSX optional product provides software emulation of these instructions, allowing you to run PDP-11 software on the MicroVAX.

Instruction emulation is completely transparent to application programs. A program produces the same results whether it is run on the MicroVAX II, with software emulation of some instructions, or on a large VAX that performs all instructions in hardware.

3 MicroVAX II Software

An **operating system** is the basic software component of any computer system. DIGITAL provides these operating systems for the MicroVAX II:

- MicroVMS, a general-purpose, multiuser operating system
- ULTRIX-32m, compatible with UNIX™ operating system (not covered in this course)
- VAXELN, for running dedicated and time-critical applications developed under VMS (not covered in this course)

This course assumes you are using the MicroVMS operating system. Like other multiuser operating systems, MicroVMS performs functions that include:

- Accepting and interpreting user commands
- Reading programs into memory and executing them
- Handling communication with peripheral devices
- Ensuring that system resources (CPU time, main memory, and mass storage) are shared among users
- Protecting each user's resources from other users

3.1 Features of the MicroVMS Operating System

MicroVMS is a specially packaged version of the VMS operating system. The operating system is supplied as a set of files; almost all of these files are identical to the operating system files on a large VAX system.

The files that make up MicroVMS are distributed on a set of 5-1/4 inch diskettes or on a single TK50 tape cartridge. When you install or upgrade MicroVMS from the distribution media, the files are placed on the fixed disk. MicroVMS is packaged modularly so you can choose the options you need. This ability to tailor the operating system is important when disk space is limited, as it is on many MicroVAX II systems.

Table 3 lists the major features of MicroVMS and shows how they are grouped into options. It is not a complete description of MicroVMS. Your MicroVMS distribution kit comes with a *Software Product Description*, which describes all the features of the operating system, and a cover letter listing the contents of each MicroVMS option.

UNIX™ is a registered trademark of AT&T.

Table 3 MicroVMS Options and Features

MicroVMS Option	Feature	Purpose
Base System (must be installed)	Basic MicroVMS system	Running programs and utilities
	DCL (DIGITAL Command Language)	Processing user commands
	DIRECTORY, COPY, RENAME, DELETE, etc.	Manipulating files
	AUTHORIZE	Maintaining user accounts
	BACKUP	Backing up files
	EDT	Creating and editing text
Common Utilities Option	HELP	Getting online information
	MAIL and PHONE	User communication
	SEARCH	Finding strings in files
	DIFFERENCES	Comparing files
	DIGITAL Standard Runoff	Formatting text
	VAX Text Processing Utility (VAXTPU)	Text editing and specialized text processing
	Error logger	Recording hardware errors detected by MicroVMS
	Miscellaneous other utilities	

Table 3 MicroVMS Options and Features (Cont)

MicroVMS Option	Feature	Purpose
Secure User Environment Option	Batch queues	Noninteractive use of the system
	Print queues	Printing files
	DISKQUOTA	Controlling use of disk space
	Access Control List editor	Setting up access control for files and other objects
	ACCOUNTING	Recording system use
Program Development Tools Option	Debugger	Locating errors in programs
	MACRO assembler	Writing assembly language programs
	Message Utility	Creating messages for programs to output
	Files needed by application programmers	
System Programming Option	MONITOR	Examining system performance
	System Dump Analyzer	Examining crash dumps
	DELTA	Debugging system programs
	Files needed by system programmers	

3.2 Optional MicroVMS Software

MicroVMS may not give your users all the capabilities they need. There are many optional software products (also called layered products) that can make your MicroVMS system more useful, such as:

- DECnet, which allows your MicroVAX/MicroVMS system to communicate with other VAX systems
- VAX Information Architecture (VIA) products, which include database management and information retrieval systems
- Programming languages including Ada (™), FORTRAN, BASIC, COBOL, Pascal, PL/I, C, and LISP
- Programmer productivity tools and subroutine packages
- Videotex software for retrieving text from local or remote databases
- Systems for producing computer-based instruction
- Office automation applications that include electronic mail, desk management, spreadsheet, word processing, graphing, and slide production
- VAX-11 RSX, for developing and running PDP-11 compatible software on your VAX
- VAXELN Toolkit, for developing dedicated applications for the MicroVAX

Your DIGITAL sales representative can help you select the DIGITAL optional products you need. Many vendors other than DIGITAL also produce software for VAX systems. You and your users can even develop your own software for the MicroVAX II, using DIGITAL programming languages and tools.

LEARNING ACTIVITY (OPTIONAL)

1. If you are not already familiar with the full range of DIGITAL software available for MicroVMS, read the chapter, **Introduction to VAX/VMS Software Products**, in the *VMS System Software Handbook*. Concentrate on the Optional VAX Software Products section.

Summary

- The MicroVAX II processor is compatible with all other processors in the VAX line.
- The MicroVAX II is available in two different cabinets:
  1. BA23 pedestal, table-top, or rack-mount
  2. BA123 floor-standing cabinet
- The front of the MicroVAX II cabinet holds the control panel; the back of the cabinet holds the I/O connection panel and CPU distribution panel.
- The four main components of the MicroVAX hardware are:
  1. The CPU
  2. Main memory
  3. Peripheral devices and controllers
  4. The Q-Bus
- The MicroVAX II architecture is a subset of the VAX architecture.
- The MicroVMS operating system is the foundation of the software of your MicroVAX/MicroVMS system.
- The MicroVMS operating system is packaged modularly. The installation kit includes:
  - The Base System (which you must install)
  - The Common Utilities option
  - The Secure User Environment option
  - The Program Development Tools option
  - The System Programming option
- MicroVMS supports many optional software products and customer-written software.

1-22.

SYSTEM START-UP AND PRODUCT INSTALLATION

Introduction

The working environment of your MicroVAX/MicroVMS system depends on the hardware and software options you choose. The MicroVMS operating system, because of its modular packaging, lets you tailor your working environment to suit a particular application.

This module presents methods for handling removable media and the steps you must take to install and start the MicroVMS operating system. It also explains how to install MicroVMS updates, MicroVMS options, and optional products that expand the range of tasks you can perform. It discusses the methods and possible reasons for shutting down the system. Finally, it explains how to remove options from the system. The sections of this module are:

- Installing the MicroVMS operating system
- Starting up the installed system
- Installing software
- Shutting down the system
- Removing MicroVMS options

Objectives

To set up a MicroVAX/MicroVMS environment, and to install optional software, you should be able to:

- Handle tape cartridges and diskettes properly:
  - Storage
  - Loading and unloading
- Install MicroVMS on the MicroVAX II from tape cartridge or diskette
- Start the system after MicroVMS has been installed
- Install MicroVMS options and other software products
- Remove MicroVMS options if more room is needed on the disk
- Shut down the system in an orderly manner

Resources

1. *MicroVMS User's Manual*, Chapter 1, Installation and Operations
2. *MicroVAX II Owner's Manual*, Chapter 2, Operations

1 Installing the MicroVMS Operating System

You should install the MicroVMS base system if you have a new MicroVAX II system or if you want to completely overwrite the contents of your system disk, superseding the old contents. If you are currently running an earlier version of the MicroVMS operating system and you want to save the user files that already exist on your system disk, you should upgrade your system to the new version rather than install the new version. (The steps for upgrading your MicroVAX/MicroVMS system are described in the *MicroVMS User's Manual*.)

1.1 Naming Conventions

Your MicroVAX II has one or more fixed disks. It may also have two diskette drives. When you install MicroVMS, you need to know the names the system assigns to the disk and diskette drives. The physical device name of a disk depends upon the number of fixed disks in your MicroVAX II system. The logical names are the same regardless of the number of disks. You should use the logical names instead of the physical device names to avoid confusion. Note that the logical names are defined only when VMS is running, not when the system is in console mode.

Table 1 lists the physical device names and logical names for disks on a MicroVAX II system.

Table 1 MicroVAX II Logical and Physical Names

Physical Device	Logical Name	Physical Name
On systems with one fixed disk:		
First fixed disk	$DISK1	DUA0
First removable diskette drive	$FLOPPY1	DUA1
First removable diskette drive	$FLOPPY2	DUA2
On systems with two fixed disks:		
First fixed disk	$DISK1	DUA0
Second fixed disk	$DISK2	DUA1
First removable diskette drive	$FLOPPY1	DUA2
First removable diskette drive	$FLOPPY2	DUA3
first tape	$TAPE1	MUA0

1.2 Storing, Handling, and Loading Diskettes and Cartridges

Before installing MicroVMS, you need to know how to handle removable media. This section describes the proper methods for storing and handling tape cartridges and floppy diskettes, and introduces the steps for loading and unloading tape cartridges and floppy diskettes. Depending on the devices on your system, you can study one or both procedures.

Table 2 lists rules for protecting tape cartridges and floppy diskettes from damage.

Table 2 Proper Handling of Removable Media

DOs	DON'Ts
Keep the volume in its protective cover when you are not using it.	Do not touch any part of the exposed tape or diskette.
Store media in a safe place away from excessive heat, magnetic fields, and steel objects.	Do not place the volume in direct sunlight or near magnetic fields.
Create a paper label and place it on your volume.	Do not bend or fold a diskette.
Use a felt-tip pen to write on a label that is attached to a floppy diskette.	Do not use a ball-point pen or a pencil to write on a label that is attached to a diskette.

1.2.1 Steps for Loading a Tape Cartridge

You load magnetic tape cartridges through the door on the front of the tape drive using the following steps. Figure 1 illustrates these steps.

❶ Set the LOAD/UNLOAD pushbutton on the tape drive to the UNLOAD (out) position.

Lift the cartridge-release handle when the red light on the tape drive is off and the green light glows steadily.

NOTE

Never lift the cartridge-release handle if the green light is not glowing. Lifting the handle may result in damage to the cartridge.

❷ Hold the tape cartridge with the volume label slot and the write-protect switch facing you. Insert the tape cartridge into the drive with the alignment arrow on top and pointing away from you.

The red light comes on when the cartridge is inserted.

❸ Lock the cartridge-release handle by pushing down.

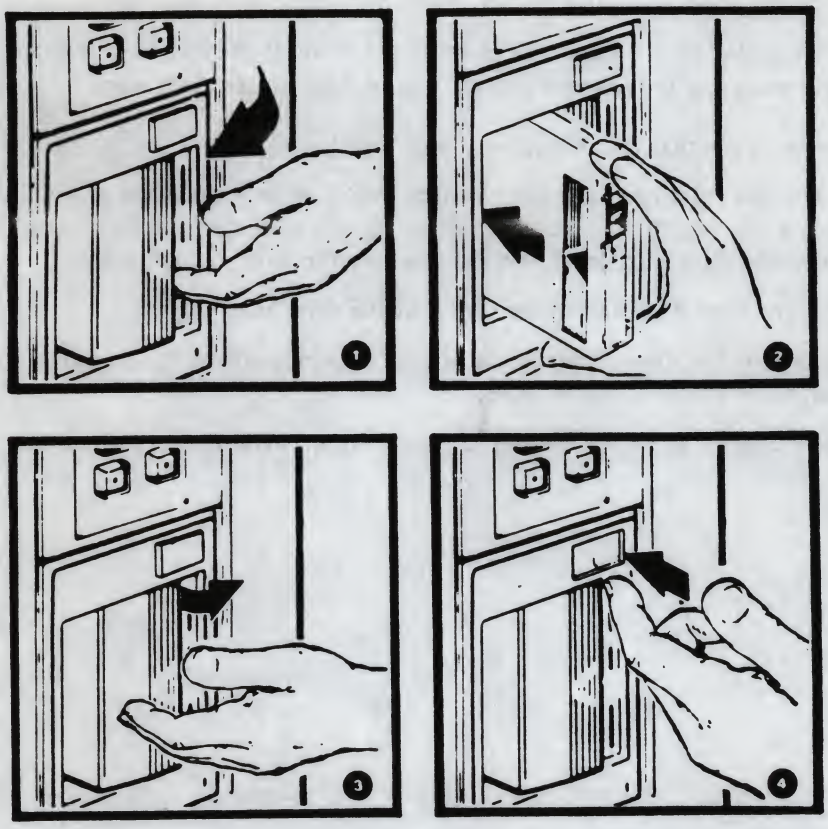
The red light goes off when the handle is locked in place.

❹ Set the LOAD/UNLOAD pushbutton to the LOAD (in) position.

As the tape is being loaded the red light glows continuously.

NOTE

Each time you use a new tape cartridge, the tape drive goes through a calibration sequence that takes approximately two minutes.



MKV85-1587

Figure 1 Loading a TK50 Tape Cartridge

1.2.2 Steps for Unloading a Tape Cartridge

1. Set the LOAD/UNLOAD switch to the UNLOAD (out) position.
2. Wait for the red light to go out and the green light to go on.
3. Lift the cartridge release handle to partly eject the cartridge.
4. Pull the tape cartridge out of the drive.
5. Lock the cartridge release handle by pushing down.

1.2.3 Steps for Loading a Floppy Diskette

You load RX50 floppy diskettes through the door on the front of the diskette drive. You can load a diskette by completing the following steps. Figure 2 illustrates these steps.

1. Check the drive to see that the red activity light is off.
2. Remove the diskette from its protective envelope, handling only the edges of the diskette. If you want to write information on the volume, be sure to remove the foil tab from the write-protect notch; otherwise, leave the foil tab over the write-protect notch.
3. Open the diskette drive door and make sure that the drive is empty.
4. Insert the diskette in the drive, keeping the orange alignment arrow on the diskette in line with the orange line on the diskette drive.
5. Close the drive door to allow the system to read from or write to the diskette.

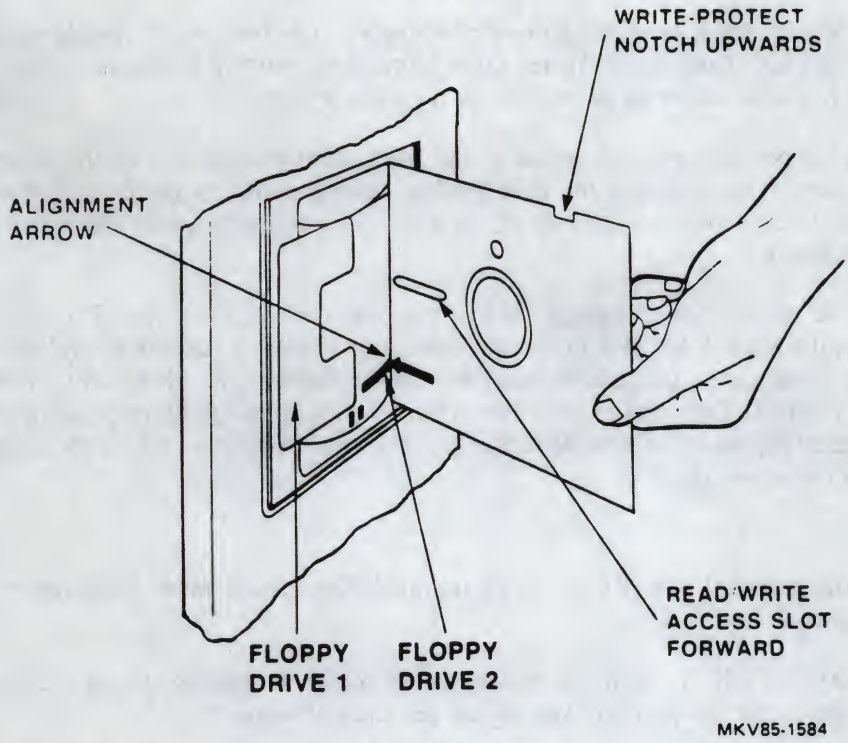


Figure 2 The RX50 Diskette Drive

1.2.4 Steps for Unloading a Floppy Diskette

1. Make sure the red activity light is off.
2. Open the diskette drive door.
3. Remove the diskette.
4. Put the diskette back in its protective envelope.
5. Close the diskette drive door.

1.3 Turning on the System for the First Time

Turning on your MicroVAX II system is a two-step operation. You turn on the console terminal and then the MicroVAX II processor. To see the informational start-up messages, turn on the console terminal and let it warm up before you turn on the processor.

After turning on the console terminal, set all of the front panel pushbuttons on the processor control panel to the out position and the Halt Enable/Disable switch to the Enable position. (The Halt Enable/Disable switch is located on the CPU distribution panel on the rear of the MicroVAX II enclosure.)

You must select the console-mode language the first time you turn on the system. This language applies only when the MicroVAX II is in console mode, not when it is running MicroVMS. To select a language, turn the language selection knob on the back of the MicroVAX II to the middle (inquire) position. Turn on the processor, then select a language by responding to the prompts. After selecting the console-mode language, save your choice by turning the language selection knob to the arrow position.

> NOTE
>
> If the console terminal is a VT100-series terminal, the console-mode language is always English.

To turn on the MicroVAX II, push the power switch to the 1 position. If the front-panel pushbuttons are set to the out position, you should see the following:

- The Ready button should glow green.
- The Write-Protect button should not glow.
- The Halt button should not glow.
- The Restart button should not glow.
- The console terminal should display a console-mode language menu.

Example 1 shows the steps for selecting North American English as the console-mode language.

```
KA630-A.V1.2
1)Dansk       7)Nederlands
2)Deutsch     8)Norsk
3)English     9)Portugues
4)Espanol     10)Suomi
5)Francais    11)Svenska
6)Italiano


(1..11): 3

1)United Kingdom
2)United States/Canada

(1..2): 2


Performing normal system tests

  7..6..5..4..3..

Tests completed.
>>>
```

Example 1 Turning on the System for the First Time

Notes on Example 1.

1. First, the system displays a list of console-mode language choices. Enter the number 3 in response to the "(1..11):" prompt. When you enter the number 3, English is selected as the default language. Finally, the system lists a second menu. Enter the number 2 to further narrow the language selection.

2. The system performs some hardware tests and begins a countdown starting with the number 7. When it reaches 3, it displays the console-mode prompt (>>>). If the console-mode prompt is not displayed, press the Halt pushbutton twice, once to set the Halt pushbutton and once to release it.

1.4 Stand-Alone BACKUP

After turning on your MicroVAX II system and selecting a console-mode language, you must load stand-alone BACKUP, which is used to copy MicroVMS from the removable media to the fixed disk. If your distribution media is a tape cartridge, stand-alone BACKUP, the base system, and all the options are contained on the same magnetic tape cartridge. If your distribution media is a set of diskettes, stand-alone BACKUP is contained on a set of floppy diskettes.

Table 3 lists the steps for loading stand-alone BACKUP.

Table 3 Loading Stand-Alone BACKUP

Step	Tape Cartridge	Diskettes
1	Load the MicroVMS distribution tape cartridge in the tape drive (MUA0).	Load the first stand-alone BACKUP diskette in the first diskette drive (DUA1 or DUA2).
2	Enter a boot command at the console prompt: `>>> B/20000 MUA0`	`>>> B DUA1` or `>>> B DUA2`
	(This command starts the system using information from the volume.)	
3	Enter the date and time at the prompt, using the 24-hour format. For example, to specify 2:15 PM on August 16, 1985 you should enter: 16-AUG-1985 14:15.	
4	(Omit this step when using a tape cartridge.)	When you are prompted, place the next diskette in drive 1. Do NOT remove the diskette when the red activity light is on.
5	(Omit this step when using a tape cartridge.)	When you have placed the next diskette in drive 1, type YES and press the RETURN key. (Stand-alone BACKUP is on 3 diskettes. The third diskette is left in the drive during the installation of the base system.)
6	The dollar sign prompt ($) indicates that stand-alone BACKUP has been loaded. You can now install the base system.	
7	Do not remove the last stand-alone BACKUP volume from the drive.	

1.5 Installing MicroVMS

The distribution kit for any version of MicroVMS comes with detailed instructions for installing it. The instructions discuss your processor, your system device, and the device where the distribution media is loaded.

If your distribution media is a tape cartridge, read Section 1.5.1; if your distribution media is floppy diskettes, read Section 1.5.2.

1.5.1 Installing MicroVMS from a Magnetic Tape Cartridge

After loading stand-alone BACKUP, leave the tape cartridge in the tape drive. This tape cartridge contains stand-alone BACKUP, the MicroVMS operating system, and the MicroVMS options. To install MicroVMS, complete the following steps:

1. Enter the following command:

   ```
   $ BACKUP/VERIFY/INITIALIZE MUA0:MICROVMS/SAVE_SET DUA0:
   ```

   This command copies the MicroVMS base system from the volume that is loaded on the MUA0 drive to the fixed disk (DUA0). The dollar sign prompt ($) indicates that the installation is complete.

2. (OPTIONAL) If you want to install options, leave the tape cartridge in the tape drive. (Installing optional software is described later in this module.)

3. Halt and restart the system:

   - Press the HALT button twice, once to set it and once to release it.
   - At the console prompt, type:

     ```
     >>> B DUA0
     ```

     Then press RETURN. This procedure is discussed in greater detail later in this module.

1.5.2 Installing MicroVMS from Floppy Diskettes

After loading stand-alone BACKUP, leave the last diskette in the diskette drive. This diskette must be available for stand-alone BACKUP to copy the MicroVMS operating system to the fixed disk (DUA0). To install MicroVMS, complete the following steps:

1. Place the first MicroVMS base system diskette in the second diskette drive (drive 2).
2. Enter the following command:

   ```
   $ BACKUP/VERIFY/INITIALIZE DUA2:MICROVMS/SAVE_SET DUA0:
   ```

   The command copies the information from the diskette in drive 2 (referred to as DUA2 or DUA3) to the fixed disk (DUA0). (Refer to Table 1 to determine the physical name for the second diskette drive.) You should see informational messages displayed on the console terminal screen. When prompted, remove the diskette from drive 2 and place the next base system diskette in the drive. After you place the diskette in drive 2, type YES and press RETURN. The copying process continues.
3. Continue replacing diskettes in drive 2 until you have processed the last diskette of the base system.
4. When the dollar sign prompt ($) is displayed, remove the diskettes from both diskette drives. The installation is complete.
5. Halt and restart the system.

2 Starting Up the Installed System

After installing the MicroVMS operating system you must halt and restart the system. To halt the system, press the Halt pushbutton twice. The first time you press the Halt pushbutton, the system is placed in console mode, the Halt pushbutton should glow red, and the console mode prompt (>>>) should be displayed. You must press the Halt pushbutton a second time to make it pop back out. At this time, the Halt pushbutton should not glow.

You can start your MicroVMS system from console mode by entering the command B DUA0 at the console mode prompt. If you set the Halt Enable/Disable switch to the Disable position, you can also start your system by pressing the Restart pushbutton. Either action starts the system using information from the system disk (the location of the operating system).

During start-up the system displays a sequence of messages. If this is the first time you are starting MicroVMS or if the system has been turned off for longer than 10 days, the system prompts you to enter the date and time. When the start-up process is complete, the system displays a message similar to the following:

```
SYSTEM job terminated at 16-SEP-1985 13:55:20.45
```

Example 2 shows a sample of the messages displayed on the console terminal during start-up.

```
❶ MicroVMS Version V4.2 24-JUN-1985 08:00

  The MicroVMS system is now executing the system startup procedure.

  The MicroVMS system is now executing the site-specific startup commands.

  %RUN-S-PROC_ID, identification of created process is 00000025

  %RUN-S-PROC_ID, identification of created process is 00000027
  %SET-I-INTSET, login interactive limit = 64, current interactive value = 0
    SYSTEM       job terminated at 16-SEP-1985 13:55:20.45
```

Example 2 Start-Up of a MicroVAX II System

Note on Example 2.

❶ When you press the Restart pushbutton or enter the B DUA0 command, the system begins the start-up operation. Each method of starting the system causes different messages to be displayed, until this message. From this message on, the start-up displays are identical.

When you see this message displayed, the system start-up procedure is complete. You can now log in.

3 Installing Software

The installation of your MicroVMS system is not complete until you have installed the mandatory update and any desired MicroVMS options. Also, you may receive periodic maintenance updates or purchase additional optional software products that are distributed on one or more removable volumes. You use the SYS$UPDATE:VMSINSTAL.COM command procedure to copy the mandatory update, MicroVMS options, or optional products (also called layered products) from the distribution media to your system disk.

During execution, VMSINSTAL loads and executes the installation procedure of the product, which controls the installation process. Since the installation procedure of each product is different, each product that you install generates different displays during installation. You should always refer to the release notes of the update or the installation guide for the product to learn how to respond to the installation displays.

3.1 Using VMSINSTAL

To use VMSINSTAL, log in to the SYSTEM account, using the console terminal, and complete the following steps:

1. Place the first diskette containing the software option in drive 1, or the MicroVMS or layered product distribution tape cartridge in the tape drive.
2. Set your default device and directory to the directory for the VMSINSTAL command procedure, as follows:

   ```
   $ SET DEFAULT SYS$UPDATE
   ```

3. Invoke the VMSINSTAL command procedure. Identify the product or option you want installed and the location of the distribution media. For example, to install the Common Utilities Option from a tape cartridge loaded on the MUA0 tape drive, enter the following:

   ```
   $ @VMSINSTAL UTIL $TAPE1
   ```

4. Reply to the prompts generated by the product's installation procedure.

   a. When asked if you are satisfied with the backup of the system disk:

      Enter YES if the disk has been backed up.

      Enter NO if you must back up files.

   NOTE

   The procedure for backing up files is discussed in Module 3.

   b. Some installation procedures prompt you for different responses. For example, the installation procedure for the System Programming Option asks you if you want to install all or only part of the option. (Refer to the installation guide of the kit you are installing for information about the prompts.)

   c. If you are installing the option from diskettes, load and unload diskettes from drive 1 as directed by the installation procedure displays. Enter YES when you have placed a new diskette in the drive.

5. When the $ prompt appears, VMSINSTAL is finished. You can now remove the distribution media from the drive.

3.2 Installing MicroVMS Updates, MicroVMS Options, and Optional Products

Once you are familiar with using the VMSINSTAL command procedure, you can use it to install MicroVMS updates, MicroVMS options, and layered products.

Example 3 describes the steps used to install the System Programmer Option of the MicroVMS operating system.

1. As soon as you have installed and started the MicroVMS base system, install the mandatory update. The mandatory update contains software that is required to update the MicroVMS operating system to the current version.

   Place the volume containing the update in the drive. Invoke the VMSINSTAL command procedure. Specify VMSMUP as the option name for the mandatory update. (See the *MicroVMS Release Notes* for more information on the mandatory update.)

   When the installation is complete, the system automatically halts. You must then put the system in console mode and restart it.

2. Next, invoke VMSINSTAL again to install the MicroVMS options. The possible MicroVMS options you can specify are:

   a. UTIL - Common Utilities Option
      (Content examples: HELP, MAIL, PHONE, SEARCH, VAXTPU, and DIGITAL Standard Runoff)

   b. USER - Secure User Environment Option
      (Content examples: Batch and print queues and tools for protecting user data)

   c. PROG - Program Development Tools Option
      (Content examples: Debugger, Linker, and MACRO assembler)

   d. SYSP - System Programming Option
      (Content examples: Monitor and Delta)

3. Again, install the mandatory update to apply necessary updates to the options you have installed.

4. Finally, install any layered products. To install layered products, invoke VMSINSTAL and enter the product name. (The product name is specified in the product's installation instructions.)

```
(1) Username: SYSTEM
    Password:
           Welcome to MicroVMS V4.2

        Last interactive login on Friday,  2-AUG-1985 16:43
        Last non-interactive login on Thursday,  1-AUG-1985 12:36
    $ SET DEFAULT SYS$UPDATE
    $ @VMSINSTAL


            VAX/VMS Software Product Installation Procedure V4.2


    It is 2-AUG-1985 at 16:47.
    Enter a question mark (?) at any time for help.

(2) * Are you satisfied with the backup of your system disk [YES]? YES

(3) * Where will the distribution volumes be mounted: DUA1:


    Enter the products to be processed from the first distribution volume set.
(4) * Products: SYSP

    The following products will be processed:

      SYSP V4.2

            Beginning installation of SYSP V4.2 at 16:49
    %VMSINSTAL-I-RESTORE, Restoring product saveset A...

(5) * Do you want to install the entire kit: YES
    %VMSINSTAL-I-RESTORE, Restoring product saveset B...
                                .
                                .
                                .
    %VMSINSTAL-I-RESTORE, Restoring product saveset J...
    %VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...

            Installation of SYSP V4.2 completed at 17:01

    Enter the products to be processed from the next distribution volume set.
(6) * Products: EXIT

            VMSINSTAL procedure done at 17:02

    $
```

Example 3 Installing the System Programmer Option

Notes on Example 3.

❶ Log in under the system manager's account (SYSTEM) and set the default directory to SYS$UPDATE. Then, enter @VMSINSTAL to start the MicroVMS standard installation procedure.

If there are other users on the system, or if your DECnet network is up and running, VMSINSTAL will display a list of the active processes, and ask if you want to continue. Answer NO, then shut down the network and tell other users to log out. This precaution ensures that no user has access to system files while VMSINSTAL is trying to update them.

❷ As a precaution, VMSINSTAL will ask if you are satisfied with the backup of your system disk. If you answer NO, the procedure will stop at this point to allow you to perform a system backup. Normally, you answer YES to this question and the procedure continues. The reason for backing up your system disk is that if the installation procedure fails, it might leave the disk in an unusable state.

❸ VMSINSTAL will ask you where the distribution volume will be mounted. For a TK50 magnetic tape cartridge, the distribution device will usually be MUA0. For an RX50 floppy diskette, the distribution device will usually be DUA1 or DUA2.

❹ If you do not specify which product you want installed when you invoke the command procedure, VMSINSTAL prompts for your reply. By entering SYSP, you request the installation of the System Programmer Option.

❺ If you enter NO, VMSINSTAL displays additional messages and accepts additional replies to determine which parts of the option to install. If you enter YES, VMSINSTAL installs all parts of the option. To save space on your system disk, you may decide not to install certain parts of the option.

❻ VMSINSTAL displays the "Products:" prompt again. If you are not installing another option, respond by entering the EXIT command and remove the distribution volume from the drive.

NOTE

You must reinstall the mandatory update after you install a MicroVMS option.

LEARNING ACTIVITIES

1. For additional information on product installation, read the following sections of the *MicroVMS User's Manual*:
   - Section 1.2.1, Installing the Base System from Diskettes
   - Section 1.2.2, Installing the Base System from Tape Cartridges
   - Section 1.2.3, Installing Options, Updates, and Products
2. Read Chapter 2, **Operations**, of the *MicroVAX II Owner's Manual.*

4 Shutting Down the System

A MicroVMS system usually remains running for long periods of time. Under certain circumstances, however, you must shut the system down. Such circumstances include:

- Performing hardware or software maintenance
- Losing electrical power
- Upgrading or updating system software

MicroVMS provides the command procedure SYS$SYSTEM:SHUTDOWN.COM for performing orderly shutdowns. Other shutdown methods are also available. For example, you can press the Halt pushbutton to suspend operation.

When you use the orderly shutdown method, MicroVMS attempts to avoid interrupting critical system operations. It warns users of the coming shutdown, and makes every attempt to preserve all system and user data. For this reason, you should use the orderly shutdown procedure, rather than the Halt pushbutton, whenever possible.

To shut down the system in an orderly manner, log in to the SYSTEM account, enter the symbol SHUTDOWN at the DCL prompt, and press RETURN. The SHUTDOWN symbol passes parameters to, and then executes, the command procedure SYS$SYSTEM:SHUTDOWN.COM. This procedure must be executed from the SYSTEM account or another suitably privileged account. Module 4 describes privileges more thoroughly.

Parameters passed to SYS$SYSTEM:SHUTDOWN.COM include: the number of minutes until shutdown, the reason for the shutdown, and when the system will be restarted. For more information on these and other parameters, refer to Section 1.4.3, Installation and Operation, in the *MicroVMS User's Manual*.

During an orderly shutdown, the SYS$SYSTEM:SHUTDOWN.COM procedure performs the following functions:

- Displays a notice on all terminals that a shutdown is imminent
- Prevents users from logging in (unless they have OPER privilege)
- Shuts down DECnet if it is installed and running
- Stops all batch and print queues
- Stops user processes
- Removes installed images
- Dismounts mounted volumes

When the shutdown is complete, press the Halt pushbutton on the control panel to put the system into console mode. Press the Halt pushbutton again to release it. Now, if you desire, restart the system from console mode.

Example 4 lists the messages that are displayed at the console terminal during shutdown.

```
(1) $ SHUTDOWN

          SHUTDOWN -- Perform an Orderly System Shutdown

%SHUTDOWN-I-OPERATOR, This terminal is now an operator's console.
%SHUTDOWN-I-DISLOGINS, Interactive logins will now be disabled.
%SET-I-INTSET, login interactive limit=0, current interactive value=1
%SHUTDOWN-I-SHUTNET, The DECnet network will now be shut down.
%SHUTDOWN-I-STOPQUEMAN, The queue manager will now be stopped.

SHUTDOWN message on SPEEDY from user SYSTEM at _SPEEDY$OPA0:  11:01:33
SPEEDY will shut down in 0 minutes; back up LATER. Please log off node
SPEEDY.
SHUTDOWN

1 terminal has been notified on SPEEDY.

%SHUTDOWN-I-STOPUSER, All user processes will now be stopped.
%SHUTDOWN-I-REMOVE, All installed images will now be removed.
%SHUTDOWN-I-DISMOUNT, All volumes will now be dismounted.
(2)          SYSTEM SHUTDOWN COMPLETE - USE CONSOLE TO HALT SYSTEM
```

Example 4 Shutting Down the System

Notes on Example 4.

(1) When you enter the SHUTDOWN symbol, the system executes the SYS$SYSTEM:SHUTDOWN command procedure. During execution, the system should display informational messages. The last message states that the system shutdown is complete.

(2) When the shutdown is complete, press the Halt pushbutton on the processor control panel to put the system into console mode. Press the Halt pushbutton again to release it. You can now restart the system from console mode, or you can turn the power off.

5 Removing MicroVMS Options

MicroVMS provides the command procedure SYS$UPDATE:REMOVE.COM for removing MicroVMS options. You should use this procedure to remove software that you no longer need. By removing unused software, you can save room on the disk.

Always use the REMOVE procedure to remove parts of the operating system. Do not try to delete operating system files yourself, as you may destroy important components of the system.

To remove an option:

1. Log in to the SYSTEM account.
2. At the $ prompt, enter the REMOVE symbol. This symbol executes the REMOVE command procedure. The procedure displays a list of the options and suboptions and prompts you for the items you want to remove.
3. Respond to the prompts generated by the procedure.

Example 5 shows the steps for removing a MicroVMS option.

```
❶ $ REMOVE

  This command procedure removes entire options or suboptions
  of the MicroVMS kit. Help may be obtained for any of the prompts
  by typing HELP.

❷ Valid options and suboptions are:

  UTIL
       !* UTIL_A *KITINSTAL.COM*
       !* UTIL_B *MAIL utility*
       !* UTIL_C *SEARCH utility*
       !* UTIL_D *DIFF utility*
       !* UTIL_E *DUMP utility*
       !* UTIL_F *RUNOFF utility*
       !* UTIL_G *PHONE utility*
       !* UTIL_H *MicroVMS HELP library*
       !* UTIL_I *Remote terminal support via SET HOST/DTE*
       !* UTIL_J *Foreign Terminal Support*
       !* UTIL_K *LAT-11 terminal server support (via Ethernet)*
       !* UTIL_L *Stand-alone backup on system disk support*
       !* UTIL_M *MicroVAX-I bootstrap that works for any MSCP system device*
       !* UTIL_N *Error Log Report Generator utility*
       !* UTIL_O *VAXTPU utility*
  USER
       !* USER_A *Default files*
       !* USER_B *File Access Control List utilities*
       !* USER_C *Disk Quota utility*
       !* USER_D *Print and Batch Queue utilities*
       !* USER_E *Input Queue Symbiont*
       !* USER_F *Accounting Log Report Generator utility*
  NET
       !* NET_A  *Default files*
       !* NET_B  *Incoming Remote File Access files*
       !* NET_C  *Incoming Remote Terminal files*
       !* NET_D  *Outgoing Remote Terminal files*
       !* NET_E  *Network Test files*
       !* NET_F  *Remote Task Loading*
```

Example 5 Removing the System Programmer Option (Sheet 1 of 3)

```
PROG
     !* PROG_A *KITINSTAL.COM*
     !* PROG_B *Debugger utility*
     !* PROG_C *Image Dump utility*
     !* PROG_D *RMS Analyze and FDL Editor utilities*
     !* PROG_E *Message utility*
     !* PROG_F *Object and Shareable Image libraries*
     !* PROG_G *Macro libraries*
     !* PROG_H *Macro assembler*
     !* PROG_I *SDL intermediary form of STARLET.MLB*
     !* PROG_J *FORTRAN require files*
SYSP
     !* SYSP_A *Default files*
     !* SYSP_B *Files-11 ODS1 ACP and EXCHANGE utility*
     !* SYSP_C *Monitor utility*
     !* SYSP_D *Analyze Object File utility*
     !* SYSP_E *Delta debugger*
     !* SYSP_F *System Dump Analyzer utility*
     !* SYSP_G *System Symbol Table file*
     !* SYSP_H *Misc Symbol Table files*
     !* SYSP_I *System map*
     !* SYSP_J *Connect-to-Interrupt Driver*

What layered VMS option(s) do you want to remove.
Options must be separated by commas, e.g., USER,UTIL,SYSP: SYSP

❸ Do you want to remove the entire SYSP option (y/n): Y
```

Example 5 Removing the System Programmer Option (Sheet 2 of 3)

Notes on Example 5.

❶ To execute the command procedure SYS$UPDATE:REMOVE.COM, enter the REMOVE symbol at the console terminal while logged in to the SYSTEM account.

❷ The command procedure displays an informational message. Then it displays the MicroVMS options and suboptions it can remove.

❸ The procedure prompts for the option you want to remove. Next, it prompts you to determine if you want all or part of the option removed. Finally, it displays an informational message for each file it deletes.

```
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSEXE]F11AACP.EXE;1 deleted (45 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSEXE]CONINTERR.EXE;1 deleted (6 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSEXE]SYS.STB;2 deleted (111 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSLIB]DELTA.OBJ;1 deleted (33 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSLIB]DELTA.EXE;1 deleted (27 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSEXE]ANALYZOBJ.EXE;1 deleted (57 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSEXE]MONITOR.EXE;1 deleted (138 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSEXE]EXCHANGE.EXE;1 deleted (120 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSEXE]SDA.EXE;1 deleted (303 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSEXE]SYS.MAP;1 deleted (771 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSEXE]SYSDEF.STB;1 deleted (138 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSHLP]EXCHNGHLP.HLB;1 deleted (135 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSHLP]INSTALHLP.HLB;4 deleted (18 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSHLP]MNRHELP.HLB;1 deleted (33 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSHLP]PATCHHELP.HLB;4 deleted (84 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSHLP]SDA.HLB;1 deleted (42 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSHLP]SYSGEN.HLB;4 deleted (66 blocks)

*****   SYSP REMOVED   *****

$
```

Example 5 Removing the System Programmer Option (Sheet 3 of 3)

Summary

- Handle tape cartridges and diskettes properly to prevent damage and loss of data.
- You must install the base system of MicroVMS before installing other software on the system. To install the base system:
  1. Turn on the console terminal and the MicroVAX II system.
  2. Load stand-alone BACKUP.
  3. Install the base system.
  4. Halt and restart the system.
- You can install other parts of the operating system to enhance your working environment. To install optional software:
  1. Log in to the SYSTEM account from the console terminal.
  2. Use VMSINSTAL to install the mandatory update.
  3. Use VMSINSTAL to install desired options.
  4. Install the mandatory update again.
  5. Restart the system.
- You must shut down the system when:
  - You have just installed new software
  - You want to turn the system off
- You can remove options from the MicroVMS operating system that you do not need.

USING REMOVABLE STORAGE MEDIA

Introduction

Removable storage media, such as tape cartridges and floppy diskettes, can serve many purposes in the MicroVMS environment. Tape cartridges and floppy diskettes can be used:

- As distribution media for layered software
- To transport data from one MicroVAX system to another
- To store backup copies of files on the fixed disk

You can use the tape cartridge or diskette drive on your MicroVAX system as an auxiliary storage device. If space is limited on the fixed disk, you can transfer files that you do not use often to a tape cartridge or floppy diskette. You can place copies of important files onto removable media for safekeeping. You can transport files to another MicroVAX system by placing files on a cartridge or a diskette, and using the removable storage media on another system.

This module introduces the characteristics of tape cartridges and floppy diskettes. It explains how to use a diskette as a work disk by creating directories and files on it. It also describes the methods for copying files to tape cartridges or floppy diskettes using the Backup utility. The sections of this module are:

- Media characteristics
- Using removable media
- Using a diskette as a work disk
- Using the Backup utility to save files
- Using the Backup utility to restore files

Objectives

To use removable storage media, you should be able to:

- Decide when to use tape cartridges or diskettes
- Prepare new tape cartridges or diskettes for use
- Create directories and files on a diskette
- Back up important files to tape cartridges or diskettes
- Restore files that have been backed up

Resource

The following resource contains information on using removable media and performing backups:

MicroVMS User's Manual, Chapter 5, Storage and Output of Data

1 Media Characteristics

With your MicroVAX II system, you can store information on either a TK50 magnetic tape cartridge or an RX50 floppy diskette. Some systems have just a tape drive or just a diskette drive, and some have both. You can use either removable media to:

- Install software, such as the MicroVMS operating system, on your MicroVAX
- Keep backup copies of files in case of damage to the fixed disk or accidental deletion
- Store files that you use infrequently
- Transport files between systems

You can also use floppy diskettes to work on the system when space on the fixed disk is low. (You cannot work directly from a tape cartridge, only a floppy diskette.)

Each media has its own characteristics. A TK50 magnetic tape cartridge:

- Stores up to 95 megabytes of information
- Has a slot on one end for a volume label
- Has a write-protect switch next to the volume label slot

An RX50 floppy diskette, which measures 5 1/4 inches square:

- Stores up to 400 kilobytes of information
- Has a paper label attached to its front side
- Has a write-protect notch on one side

Each of these media has its own method of protecting information from being overwritten. With a tape cartridge, the system can write information on the volume only when the write-protect switch is positioned to the right. With an RX50 diskette, the system can write data on the volume only when the write-protect notch is exposed. To write-protect a diskette, place a foil tab over the notch.

Figure 1 illustrates a TK50 magnetic tape cartridge. Figure 2 illustrates an RX50 floppy diskette.

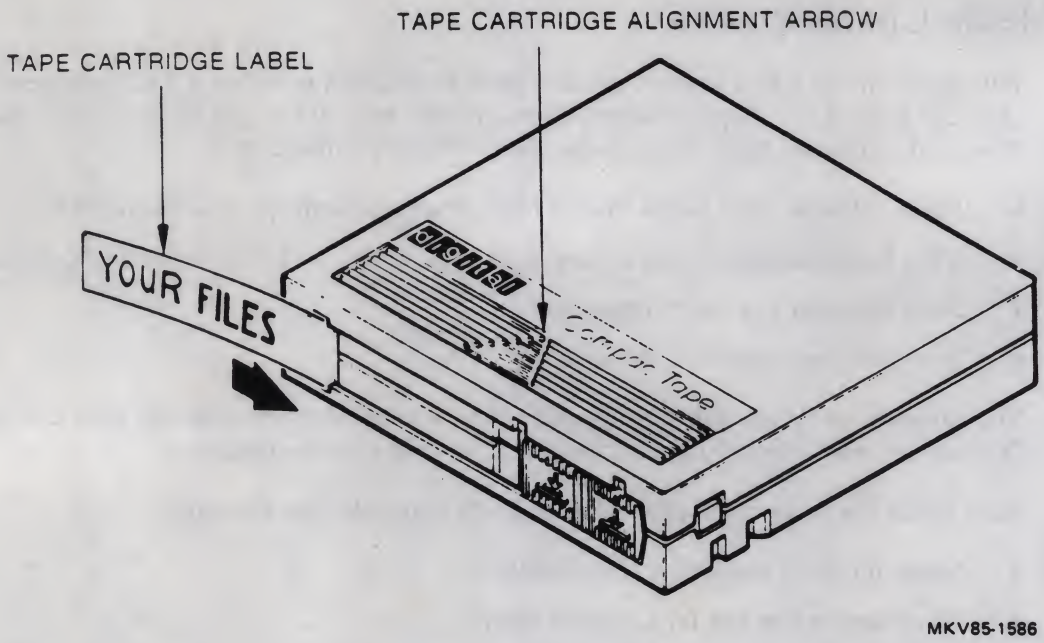


Figure 1 A TK50 Magnetic Tape Cartridge

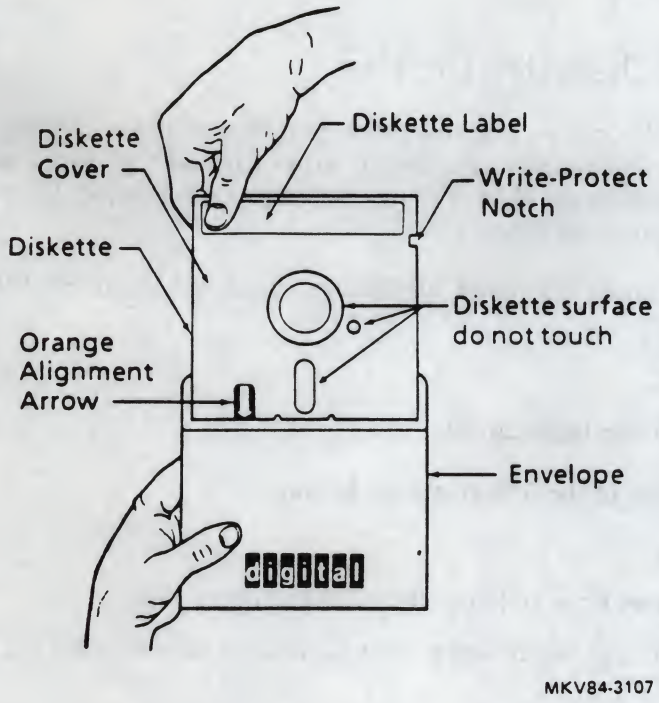


Figure 2 An RX50 Floppy Diskette

2 Using Removable Media

2.1 Preparing Tapes and Diskettes for Use

Before storing data on a new diskette or tape cartridge, you must initialize it using the INITIALIZE command. During initialization, the system writes data such as owner name, volume protection, and volume label on the diskette or tape cartridge. On a diskette, the system also creates system files used to maintain file structures.

Since initialization destroys any previously stored information, you should initialize a volume only if:

- It is a new volume, or
- It contains information that can be discarded

To initialize a volume, at least one of the following must be true:

- The volume is blank (new).
- The owner User Identification Code (UIC) of the volume matches yours.
- You have the VOLPRO privilege, which is required to initialize a volume with a UIC that is different from yours.

NOTE

Module 4 discusses the method for setting privileges.

To initialize a volume, load it into the drive and then issue the INITIALIZE command. The INITIALIZE command format is as follows:

```
$ INITIALIZE device-name volume-label
```

device-name

The device name specifies the device that contains the volume to be initialized. You can specify either the physical device name or the system-defined logical name for the device. DIGITAL recommends that you use the system-defined logical name. Following is a list of the system-defined logical names for the diskette and tape cartridge drives.

- $TAPE1 – Logical name for the TK50 tape drive
- $FLOPPY1 – Logical name for the first diskette drive
- $FLOPPY2 – Logical name for the second diskette drive

volume-label

The volume label specifies the identification to be placed on the volume. For a disk volume, you can specify a maximum of 12 alphanumeric characters. For a tape volume, you can specify a maximum of 6 alphanumeric characters.

NOTE

You should always write the label of the volume on a paper label and place it on the diskette or in the slot at the end of the tape cartridge. You will need to know the volume label when you use the volume again.

Following are two INITIALIZE command examples.

```
$ INITIALIZE $TAPE1 MYVOL01

$ INITIALIZE $FLOPPY1 MYVOL01
```

2.2 Mounting and Dismounting a Removable Volume

After you have initialized a new volume (or a volume whose entire previous contents you wish to discard), or have simply loaded an old volume, you must mount it using the MOUNT command. When mounted, the device becomes unavailable to other users. The MOUNT command format is as follows:

```
$ MOUNT[/FOREIGN] device-name volume-label [logical-name]
```

[/FOREIGN]

The /FOREIGN qualifier is optional. You must omit the /FOREIGN qualifier to use a diskette as a work disk.

You must include the /FOREIGN qualifier with the MOUNT command to save or restore files using the Backup utility. The /FOREIGN qualifier informs the system that the information stored on the volume has a format other than the one MicroVMS uses. The Backup utility processes files in a method different from the normal file system.

If you use the /FOREIGN qualifier, the only parameter you must include is the device name.

device-name

The device name specifies the device that contains the initialized volume to be mounted. As in the INITIALIZE command, you can specify either the physical device name or the system-defined logical name for the device.

volume-label

The volume label is the string of alphanumeric characters that you assigned to the volume during its initialization.

[logical-name]

Logical-name is an optional parameter. If entered, the system creates a user-defined logical name for the device. You may choose to include this parameter to create an easy-to-remember synonym for the device name.

Following are two examples of the MOUNT command:

```
$ MOUNT $TAPE1 MYVOL01 MYTAPE

$ MOUNT $FLOPPY1 MYVOL01 MYDISK
```

When you are finished reading from and writing to the removable volume, you must enter the DISMOUNT command before removing it from the drive. The DISMOUNT command format is as follows:

```
$ DISMOUNT[/NOUNLOAD] device-name
```

When you enter the DISMOUNT command, the system frees the device for use by another user. You can enter either the physical device name or a logical name for the device as a parameter to the DISMOUNT command. If you are using a tape volume, you can prevent the tape from being unloaded during the dismount operation by including the /NOUNLOAD qualifier with the DISMOUNT command.

NOTE

A mounted device is dismounted automatically when you log out or when you shut down the system.

Following are two examples of the DISMOUNT command:

```
$ DISMOUNT $TAPE1

$ DISMOUNT $FLOPPY1
```

Example 1 shows the steps for preparing, mounting, and dismounting a volume.

```
❶ $ INITIALIZE $FLOPPY1 MYVOL01
  $
❷ $ MOUNT $FLOPPY1 MYVOL01 MYDISK
  %MOUNT-I-MOUNTED, MYVOL01      mounted on _DUA1:
❸ $ DIRECTORY MYDISK:[000000]

  Directory MYDISK:[000000]

  000000.DIR;1      BACKUP.SYS;1      BADBLK.SYS;1      BADLOG.SYS;1
  BITMAP.SYS;1      CONTIN.SYS;1      CORIMG.SYS;1      INDEXF.SYS;1
  VOLSET.SYS;1

  Total of 9 files.
  $
❹ $ DISMOUNT MYDISK
  $
```

Example 1 Preparing, Mounting, and Dismounting a Diskette

Notes on Example 1.

❶ During initialization, the system writes the volume label MYVOL01 on the floppy diskette loaded into $FLOPPY1. Also, the system creates system files on the diskette. The system uses these files to maintain the diskette's file structure.

❷ The MOUNT command makes the volume available to you for processing and unavailable to other users. This command mounts the volume MYVOL01 on the device $FLOPPY1. Also, it assigns the additional logical name MYDISK to $FLOPPY1. ($FLOPPY1 is the system-defined logical name for the diskette drive.)

❸ Since the volume is mounted without the /FOREIGN qualifier, the system treats the disk like a normal file-structured volume. This DIRECTORY command lists the system files created during initialization. (You get the same results if you substitute $FLOPPY1 for the logical name MYDISK.) Never modify or delete any of these files.

❹ The DISMOUNT command makes the device available to any user on the system, and deletes the logical name MYDISK.

3 Using a Diskette as a Work Disk

Once you have initialized a diskette you can mount it, create directories on it, and enter commands to create and modify files on it. (You cannot create directories or perform certain other operations on a tape cartridge.) The steps for using DCL commands to manipulate files on a diskette are as follows:

1. Initialize the diskette, if necessary.
2. Mount the diskette.
3. Create a directory on the diskette.
4. Set your default directory to the diskette directory.
5. Complete the tasks you wish to perform.
6. Dismount the diskette when you are finished.

Example 2 illustrates the steps for using DCL commands to manipulate files on a diskette.

```
❶ $ MOUNT $FLOPPY1 MYVOL01 MYDISK
%MOUNT-I-MOUNTED, MYVOL01      mounted on _DUA1:
$
❷ $ CREATE/DIRECTORY MYDISK:[FILES]
$
❸ $ SET DEFAULT MYDISK:[FILES]
$
❹ $ COPY $DISK1:[SMITH]FILE*.TXT *.*
$
$ DIRECTORY

Directory MYDISK:[FILES]

FILE1.TXT;2          FILE2.TXT;1

Total of 2 files.
$
❺ $ DISMOUNT MYDISK
$
$ SET DEFAULT $DISK1:[SMITH]
$
```

Example 2 Working with Diskettes

Notes on Example 2.

❶ This MOUNT command makes the MYVOL01 volume available on the device $FLOPPY1. Also, it assigns the MYDISK logical name to the device.

❷ The CREATE/DIRECTORY command creates the FILES directory in the master file directory on the diskette.

❸ This command establishes the FILES directory on the diskette as your default directory.

❹ The COPY command moves copies of files from the SMITH directory on $DISK1, the fixed disk, to the FILES directory on the diskette. The highest version of each file in the SMITH directory that begins with the "FILE" string is copied.

❺ When you are done using the diskette, dismount the volume. Use the SET DEFAULT command to change your default directory back to your login directory on the fixed disk.

4 Using the Backup Utility

Although you can use the COPY command to save copies of files, you can protect your data more effectively by saving it with the Backup utility. You should back up your files regularly to protect against failure of the hard disk or accidental deletion of files.

Like the COPY command, the Backup utility can:

- Transfer files to a tape or a diskette
  - Single files
  - Entire directory structures NIET !

In addition, you can use the Backup utility to:

- Save files in a special file called a save set.
- Create a report listing information about a save set.
- Restore files stored in a save set.

You should use the Backup utility instead of the COPY command for making backup copies of files. The Backup utility:

- Has better error correction than the COPY command.
- Works faster than the COPY command.
- Gives you more command options.
- Generates informative reports.

4.1 Using the Backup Utility to Produce a Save Set

You can use the Backup utility to create a special file called a **save set**, which contains all the files you specify in the BACKUP command. To store files in a save set, follow these steps:

1. Mount a removable volume using the /FOREIGN qualifier.
2. Enter the BACKUP command to create the save set containing the files you specify.
3. Dismount the volume.

Examples 3 and 4 list the commands for backing up a file to a save set and a directory structure to a save set, respectively.

```
$ MOUNT/FOREIGN $TAPE1
(1) %MOUNT-I-MOUNTED, MYVOL01      mounted on _MUA0:
$
(2) $ BACKUP TEXTFILE.TXT
_To: $TAPE1:MY_SAVE_SET.SAV/SAVE_SET
$
(3) $ DISMOUNT $TAPE1
$
```

Example 3 Backing Up a File to a Save Set on Tape

Notes on Example 3.

(1) After you enter the MOUNT command, the system responds with an informational message. The message identifies that the volume MYVOL01 is mounted on the device MUA0 (the physical device name of the TK50 tape drive).

(2) **TEXTFILE.TXT** – You are backing up a single file, TEXTFILE.TXT.

$TAPE1:MY_SAVE_SET.SAV – This output specification specifies the save set. It does not contain a directory name, because the volume is mounted with the /FOREIGN qualifier and has no directory structure. On a diskette, the system places the save set in the master file directory ([000000]).

The BACKUP command moves the file TEXTFILE.TXT into the save set MY_SAVE_SET.SAV on the volume mounted in the tape drive $TAPE1.

/SAVE_SET – Specifies that the file is a save set. To specify a disk file as a save set, you must include this qualifier.

To create a save set on a tape cartridge, the /SAVE_SET qualifier is optional. The Backup utility assumes that a file stored on tape is a save set.

(3) By dismounting the device, you make it available to another user.

```
$ INITIALIZE $TAPE1 MYVOL01
$ MOUNT/FOREIGN $TAPE1
%MOUNT-I-MOUNTED, MYVOL01       mounted on _MUA0:
$
$ BACKUP [SMITH...]*.*;* $TAPE1:MINE.SAV/SAVE_SET
$
$ DISMOUNT $TAPE1
$
```

Example 4 Backing Up a Directory Structure to a Save Set

Note on Example 4.

1. In this example, the MINE.SAV save set contains the file structure from the directory SMITH on the default device. Wildcards are allowed with the BACKUP command.

Table 1 lists common input file specifications used when backing up files. Table 2 lists common qualifiers used when backing up files.

Table 1 Common Input File Specifications Used when Backing Up Files

Files to be Backed Up	Specification Example
The most recent version of a file	`MYFILE.TXT;`
All versions of all files in a directory	`[SMITH]*.*;*`
All versions of all files in the entire directory structure	`[SMITH...]*.*;*`

Table 2 Common Qualifiers Used when Backing Up Files

Function	Qualifier
Compares the files in the save set to the original files	/VERIFY
Deletes the original files after backing them up	/DELETE
Displays the file specification of each file processed during your backup session	/LOG
Creates a text file listing of the files processed during the backup session	/LIST=file-spec
Backs up files even if they are open (being used by other users) By default, open files are not backed up	/IGNORE=INTERLOCK

4.2 Save Sets that Span Multiple Volumes

You may want to back up more files than can fit on a single volume, especially if the volume is a diskette. You can create a multiple-volume save set by:

1. Enabling the LOG_IO or PHY_IO privilege. (Use the SET PROCESS/PRIVILEGE command.)
2. Mounting a volume using the /FOREIGN qualifier.
3. Entering the BACKUP command.
4. Responding to the prompts, by loading and unloading successive volumes.
5. Dismounting the last volume.

```
❶ $ SET PROCESS/PRIVILEGE=LOG_IO
  $
❷ $ MOUNT/FOREIGN $FLOPPY1 MYVOL01 MYDISK
  %MOUNT-I-MOUNTED, MYVOL01      mounted on _DUA1:
  $
❸ $ BACKUP [SMITH...] MYDISK:ALL.SAV/SAVE_SET
  %BACKUP-I-RESUME, resuming operation on volume 2
  %BACKUP-I-READYWRITE, mount volume 2 on _DUA1: for writing
❹ Enter "YES" when ready: YES
  $
❺ $ DISMOUNT MYDISK
```

Example 5 Creating a Multiple Volume Disk Save Set

Notes on Example 5.

❶ Creating a multiple volume save set requires LOG_IO or PHY_IO privilege. Module 4 discusses setting privileges in greater detail.

❷ To use the Backup utility to save files, you must mount the volume using the /FOREIGN qualifier. This command mounts the volume MYVOL01 on the first diskette drive, which has the logical name $FLOPPY1 assigned to it. It also assigns the additional logical name MYDISK to the device.

❸ On a diskette, the system places the save set in the master file directory ([000000]).

Including the /SAVE_SET qualifier is not optional, since the save set is to be created on a disk. This qualifier is the default when you refer to a tape file.

❹ When the system runs out of room on the diskette, it prompts you to place an additional floppy diskette in the drive. After placing the next volume in the drive, enter YES as requested by the terminal display. The system initializes the next volume and continues the operation. In this case, the system assigns the label MYVOL02 to the second volume.

❺ When the $ prompt is displayed on your terminal, enter the DISMOUNT command. This command makes the drive available to other users.

4.3 Listing the Contents of a Save Set

To find out the name of the save set(s) on a removable volume, you can issue a DIRECTORY command for the volume after you have mounted it. (Do not use the /FOREIGN qualifier when you mount the volume.) After deciding which save set to list, enter the BACKUP command with the /LIST qualifier. (This is one case in which you can use the Backup utility to get access to files on a volume that is not mounted with the /FOREIGN qualifier.) The system displays a report about the contents of the save set. Finally, dismount the volume.

Example 6 introduces the steps for listing the contents of a save set.

```
❶ $ MOUNT/OVERRIDE=IDENTIFICATION $FLOPPY1
  %MOUNT-I-MOUNTED, MYVOL01       mounted on _DUA1:
❷ $ DIRECTORY $FLOPPY1:[000000]*.SAV

  Directory $FLOPPY1:[000000]

  DIR.SAV;1

  Total of 1 files.
❸ $ BACKUP/LIST $FLOPPY1:[000000]DIR.SAV/SAVE_SET
  Listing of save set

  Save set:           DIR.SAV
  Written by:         SMITH
  UIC:                [200,001]
  Date:               20-DEC-1984 15:57:53.23
  Command:            BACKUP [SMITH...] $FLOPPY1:DIR.SAV/SAVE_SET
  Operating system:   VAX/VMS version V4.2
  BACKUP version:     V4.2
  CPU ID register:    07000401
  Node name:          _SPEEDY::
  Written on:         _SPEEDY$DUA1:
  Block size:         32256
  Group size:         10
  Buffer count:       3

❹ [SMITH]CHAPTER1.TXT;1                          1  11-DEC-1984 12:37
  [SMITH]DOCUMENTS.DIR;1                         1  14-DEC-1984 15:57
  [SMITH.DOCUMENTS]MEMO.TXT;1                    1   9-NOV-1984 12:56
  [SMITH.DOCUMENTS]MYFILE.TXT;1                  1  19-NOV-1984 15:58
  [SMITH]LOGIN.COM;2                             1   9-NOV-1984 12:47
  [SMITH]MAIL.MAI;1                             29   8-NOV-1984 15:06
  [SMITH]PROGRAMS.DIR;1                          1  14-DEC-1984 15:57
  [SMITH.PROGRAMS]CALC.PAS;1                     1   9-NOV-1984 12:56
  [SMITH]SETHOST.LOG;1                           1  14-DEC-1984 15:37
  [SMITH]TEXTFILE.TXT;1                          1  11-DEC-1984 12:35

  Total of 10 files, 38 blocks
  End of save set
  $ DISMOUNT $FLOPPY1
  $
```

Example 6 Listing the Contents of a Save Set

Notes on Example 6.

1. If you did not attach a label to the volume and you cannot remember the label of a volume that you own, you can still mount it by entering the /OVERRIDE qualifier with the MOUNT command.

   If you remember the volume label, mount it as follows:

   ```
   $ MOUNT $FLOPPY1 MYVOL01
   ```

2. This command lists all files in the top-level directory on the diskette that have the .SAV file type. The Backup utility always places save sets in the top-level directory of a disk mounted with the /FOREIGN qualifier.

3. This BACKUP command displays information about the DIR.SAV save set. Notice that the top-level directory is specified in this command. If the /FOREIGN qualifier is used with the MOUNT command, you do not include the directory name in the save set file specification.

4. For each file, the directory and file name, size in blocks, and creation date are displayed.

NOTE

You can complete the same steps shown in Example 6 using a tape cartridge. When using a tape cartridge, you specify $TAPE1 rather than $FLOPPY1 and you do not specify [000000], since directory structures do not exist on tapes.

5 Using the BACKUP Command to Restore Files

Once you have files backed up, you may find the need to restore one or more of the files. For example, you may want to restore a file that you accidentally deleted from the fixed disk or a file that you use infrequently. To restore files from a save set:

1. Mount the volume using the /FOREIGN qualifier.
2. Issue the BACKUP command to restore the files.
3. Dismount the volume.

Example 7 lists the steps for restoring files from a save set.

```
$ SHOW DEFAULT
  DUA1:[SMITH]
$
$ DIRECTORY [SMITH...]
%DIRECT-W-NOFILES, no files found
$
❶ $ MOUNT/FOREIGN $FLOPPY1
%MOUNT-I-MOUNTED, MYVOL01      mounted on _DUA1:
$
❷ $ BACKUP $FLOPPY1:DIR.SAV/SAVE_SET [*...]
$
$ DISMOUNT $FLOPPY1
❸ $ DIRECTORY [SMITH...]

Directory $DISK1:[SMITH]

CHAPTER1.TXT;1    DOCUMENTS.DIR;1    LOGIN.COM;2      MAIL.MAI;1
PROGRAMS.DIR;1    SETHOST.LOG;1      TEXTFILE.TXT;1

Total of 7 files.

Directory $DISK1:[SMITH.DOCUMENTS]

MEMO.TXT;1          MYFILE.TXT;1

Total of 2 files.

Directory $DISK1:[SMITH.PROGRAMS]

CALC.PAS;1

Total of 1 file.

Grand total of 3 directories, 10 files.
$
```

Example 7 Restoring Files

Notes on Example 7.

❶ First, Smith issues a SHOW DEFAULT command to ensure that the correct default directory is set. Then, Smith enters the MOUNT command with the /FOREIGN qualifier to make the volume MYVOL01 available for processing by the Backup utility. (The device name is the only parameter that must be entered when you include the /FOREIGN qualifier with the MOUNT command.)

❷ This BACKUP command directs the Backup utility to restore all the files that are in the DIR.SAV save set. Copies of the files in the save set are created on the default device $DISK1 in the original directory structure. (If the save set spans multiple volumes, the system issues a prompt when it is ready for the next volume.)

To place all the files from the save set into the same directory, specify the directory explicitly. For example, to place all the files into the subdirectory SUB1 under the directory SMITH, substitute [SMITH.SUB1] for [*...].

❸ This DIRECTORY command displays the restored files. The directory structure restored is the same as the structure presented in the save set listing in Example 6.

Table 3 lists some of the qualifiers to the BACKUP command that you can use to restore files.

Table 3 Common Qualifiers Used when Restoring Files

Function	Qualifier
Create new versions of the files	/NEW_VERSION
Replace existing files with files from the save set	/REPLACE
Specify files to restore from the save set	/SELECT=(file-spec)
Verify that files are properly restored	/VERIFY

Summary

- To prepare a removable volume for use, you must:
  1. Initialize it, if it is new or if you want to discard its contents.
  2. Mount it, to make it known to the system.
- You should use a diskette as a work disk if space on the fixed disk is low, or if you want to transport files to another system. To work using diskettes:
  1. Create a directory for yourself on the diskette.
  2. Set your default to your directory on the diskette.
  3. Complete the tasks you wish to perform.
  4. Dismount the diskette when you are finished.
- The Backup utility can place copies of files in special files called save sets.
- Use the Backup utility to:
  - Store important files on a removable volume.
  - List information about files stored in save sets.
  - Restore files that have been stored in a save set.
- To use the Backup utility:
  1. Mount the volume using the /FOREIGN qualifier.
  2. Issue the BACKUP command to create the save set.
  3. If necessary, load the subsequent volumes.
  4. Dismount the last volume.

SYSTEM MANAGEMENT FUNCTIONS

Introduction

The person responsible for maintaining a MicroVAX/MicroVMS system is called the system manager. On a single-user system, you are your own system manager. If there are multiple users, one user can be system manager or several users can share system management responsibilities.

As system manager, you create user accounts and the directories that will hold user files. This module discusses how to create, modify, and delete user accounts.

You also need to keep backup copies of all files on the system. This module discusses how to make regular backups of all the user files on the system, and how to restore files if a disk is destroyed.

You should know how to find information about the status of the system. This information is especially useful if problems occur. This module shows you how to use some of the informational utilities provided with MicroVMS.

Objectives

To properly manage a MicroVAX/MicroVMS system, you should be able to:

- Set the privileges necessary for certain operations
- Identify the system files that contain information about user accounts
- Create, modify, and delete user accounts
- Set up a terminal for automatic login
- Maintain the system-specific start-up command procedure
- Maintain a system-wide login command procedure
- Gain access to the system when normal login is impossible
- Set up and use batch and print queues
- Get information about system status
- Stop processes on the system
- Back up all user files on a regular basis
- Restore a disk volume if it has been destroyed

Resources

The following resources contain information on the topics in this module:

1. *MicroVMS User's Manual*, Chapter 1, Installation and Operations
2. *MicroVMS User's Manual*, Chapter 2, Accounts and Security
3. *MicroVMS User's Manual*, Chapter 5, Storage and Output of Data

1 Using a Privileged Account

Each user account has certain **privileges** associated with it. Privileges determine which DCL commands and other operations the user can execute. Because some commands affect the entire system, most users should be prevented from using them. Most user accounts need only a few privileges. The system manager should be able to issue commands that affect the system, and therefore needs an account with more than the normal privileges.

Typically, TMPMBX and NETMBX are the only privileges assigned to a user account. The TMPMBX privilege allows a user to use the SUBMIT and PRINT commands (discussed later in this module) and the SPAWN command. The NETMBX privilege allows the use of DECnet. An account with only these privileges is called a **nonprivileged account**.

Additional privileges should be given only to experienced users who need them. For example, the SYSPRV privilege allows you to do many of the system management operations; the VOLPRO privilege allows a user to override volume protection. An account with more than the normal privileges is called a **privileged account**.

Table 1 defines some of the privileges you can assign to an account. For a complete list of privileges, refer to the table in Section 2.1.1.6, Privileges, of the *MicroVMS User's Manual*.

Table 1 Privileges

Use	Privilege	Operations Permitted
Using the system	TMPMBX	Allows you to use SUBMIT, PRINT, SPAWN
	NETMBX	Allows you to communicate over a network
	VOLPRO	Allows you to override volume protection
Managing the system	SYSPRV	Gives you SYSTEM access to files
	GROUP	Allows you to affect any process in your UIC group, including stopping the process
	WORLD	Allows you to affect any process on the system
	READALL	Gives you read access to any file
	SETPRV	Allows you to set any privilege

The SYSTEM account is present on every MicroVMS system. Since the SYSTEM account has all privileges enabled, you are advised to use it only for software installation. Other system management duties can be done from the USERP account, also supplied with MicroVMS. USERP is authorized to enable any privilege, but has only TMPMBX and NETMBX enabled by default.

To enable privileges, use the SET PROCESS/PRIVILEGE command. The system displays a message at your terminal if you try to enable a privilege you are not authorized to use. Example 1 shows how to use the SET PROCESS command to enable and disable privileges.

```
❶ $ SET PROCESS /PRIVILEGE=SYSPRV
  $
❷ $ SET PROCESS /PRIVILEGE=(WORLD,SYSPRV)
  $
❸ $ SET PROCESS /PRIVILEGE=(NOWORLD,NOSYSPRV)
  $
❹ $ SET PROCESS /PRIVILEGE=(ALL,NOBYPASS)
  $
```

Example 1 Enabling and Disabling Privileges

Notes on Example 1.

❶ The SYSPRV privilege is needed to perform many system management functions.

Since it permits you to gain system access to files, you can run the Authorize utility and modify user accounts.

❷ You can enable several privileges with a single SET PROCESS command. To enable more than one privilege with the same command, separate each privilege with a comma and enclose the list of privileges in parentheses.

❸ After completing a privileged task, you should disable the privilege. To disable a privilege, add the "NO" prefix to the privilege.

❹ ALL is a synonym for all privileges. You may want to enable every privilege when you are not sure which privileges you need for a task.

NOTE

In this example, every privilege except BYPASS is enabled. You should only enable the BYPASS privilege when absolutely necessary, because it gives you unrestricted access to all files. Making a typing mistake with the BYPASS privilege enabled can be disastrous.

2 Managing User Accounts

The User Authorization File (UAF) stores information about user accounts. The UAF is the file SYS$SYSTEM:SYSUAF.DAT. Every user needs an account entry in the UAF and a login directory. The system uses the account entry to set up the user's working environment each time the user logs in. The user uses the login directory to store files.

Each time you log in, MicroVMS reads the record corresponding to your user name from the user authorization file. The system creates an interactive session for you (if you entered the correct password) with the limits and restrictions found in the various fields of your UAF record.

To manage user accounts, you use:

- The Authorize utility, which manages the UAF
- The command procedure SYS$MANAGER:ADDUSER.COM, which uses the Authorize utility to create accounts

2.1 Creating a User Account

To create a new account, you must:

1. Log in to an account and enable the SYSPRV privilege.
2. Execute the command procedure SYS$MANAGER:ADDUSER.COM.
3. Respond to the prompts. ADDUSER.COM adds a new record to the UAF and a new user directory to a fixed disk.

Example 2 shows the steps for adding an account for Jane Doe, using the ADDUSER command procedure.

```
1  $ SET PROCESS /PRIVILEGE=SYSPRV
2  $ @SYS$MANAGER:ADDUSER

3  Username(s) - separate by commas: JANE

    *** Processing JANE's account ***

4  Full name for JANE: JANE DOE
5  Password (password is not echoed to terminal) [JANE]:
6  UIC Group number (enter ? to list all UIC's) [200]:
7  UIC Member number (enter ? to list UIC  members) []: ?

   Owner        Username    UIC      Account  Privs Pri Directory

                DEFAULT   [200,200]           Normal  4 $DISK1:[USER]
                USERP     [200,200]           All     4 $DISK1:[USER]
                USER      [200,200]           Normal  4 $DISK1:[USER]
   JOHN DOE     JOHN      [200,001]  FINANCE  All     4 $DISK1:[JON]

   UIC Member number (enter ? to list UIC  members) []: 002

8  Account name []: FINANCE
9  Privileges [TMPMBX,NETMBX]:

10 Login directory [JANE]:
   Login device [$DISK1:]:

   %CREATE-I-CREATED, $DISK1:[JANE] created
   %UAF-I-ADDMSG, user record successfully added
   %UAF-I-RDBADDMSGU, identifier JANE value:
     [000200,000002] added to RIGHTSLIST.DAT
   %UAF-I-DONEMSG, system authorization file modified
   %UAF-I-RDBDONEMSG, rights database modified
```

Example 2 Adding a New User (Sheet 1 of 2)

Notes on Example 2.

1. You must have the SYSPRV privilege enabled to add a new account, because you need system access to the user authorization file.

2. This command executes ADDUSER.COM, which is in the SYS$MANAGER directory.

3. Normally, the user name is the first or last name of the individual who will be using the account, or perhaps the name of a project associated with the account. In this case, Jane Doe's first name is selected for the user name of the account. You must select a unique user name for each account on the system.

4. The full name is kept in the UAF only to help you remember who uses the account. Your response to this prompt does not have to be unique. One user can own multiple accounts.

5. When you press the RETURN key without specifying a password, you select the default value for the password. The default value is enclosed in brackets ([]). In this case, "JANE" is selected as the initial password for the JANE account. (The ADDUSER procedure always chooses the user name as the default value for the password.)

   The first time the JANE account is used, the system displays the following message:

   ```
   WARNING - Your password has expired; update immediately with SET PASSWORD!
   ```

6. A User Identification Code (UIC) identifies your account. The code contains two numbers. The first number is the group number, and the second is the member number.

   When you press the RETURN key without specifying a value, the group number for the account is assigned the octal (base eight) value 200. The default values for this and many other fields in the UAF record are provided by the DEFAULT account, which is shipped with your MicroVMS system.

   You should assign the same group number to the accounts of individuals who work on related projects. Users with the same group number can share files that are protected from access by other users.

```
(11) Check newly created account:

Username: JANE                  Owner:  JANE DOE
Account:  FINANCE               UIC:    [200,002] ([FINANCE,JANE])
CLI:      DCL                   Tables: DCLTABLES
Default:  $DISK1:[JANE]
LGICMD:   LOGIN
Login Flags:
Primary days:   Mon Tue Wed Thu Fri
Secondary days:                      Sat Sun
No access restrictions
Expiration:      (none)  Pwdminimum:  6   Login Fails:     0
Pwdlifetime: 180 00:00   Pwdchange:      (pre-expired)
Last Login:      (none) (interactive), (none) (non-interactive)
Maxjobs:           0  Fillm:         20  Bytlm:          4096
Maxacctjobs:       0  Shrfillm:       0  Pbytlm:            0
Maxdetach:         0  BIOlm:         18  JTquota:        1024
Prclm:             2  DIOlm:         18  WSdef:           300
Prio:              4  ASTlm:         24  WSquo:           512
Queprio:           0  TQElm:         10  WSextent:       1024
CPU:         (none)   Enqlm:         30  Pgflquo:       10000
Authorized Privileges:
  TMPMBX NETMBX
Default Privileges:
  TMPMBX NETMBX
%UAF-I-NOMODS, no modifications made to system authorization file
%UAF-I-RDBNOMODS, no modifications made to rights database

(12) Is everything satisfactory with the account [YES]:

$
```

Example 2 Adding a New User (Sheet 2 of 2)

Notes on Example 2 (Cont).

❼ The member number assigned to an account should be unique within the UIC group. To determine which member numbers are in use, enter a question mark (?).

❽ The account name is an optional identifier that can be used for accounting of system resources. Do not confuse it with the user name for the account. There is no default value for the account name. You can supply a value of up to 8 characters.

❾ Typically, you assign privileges to accounts as they are needed. If you press the RETURN key without specifying any privileges, only the TMPMBX and NETMBX privileges are assigned to the JANE account.

❿ By selecting the default values for the login directory and login device, you direct the system to create the JANE directory on the device associated with the $DISK1 system logical name.

⓫ After you select a login directory and device, the system creates the user directory and adds a record to the UAF. It then displays the values from the record it just created.

⓬ If you press the RETURN key, the system returns you to DCL. If you enter NO, the system deletes the JANE directory and the UAF record, and prompts you to reenter values.

2.2 Displaying and Modifying a User Account

You can use the Authorize utility to modify, copy, or remove currently existing UAF records. (You can also use the Authorize utility to add new records, but it does not automatically perform some of the tasks that ADDUSER.COM performs.)

There are many reasons for modifying an existing account. For example, you might be required to:

- Change the password of an account, because a user forgot the password or did not change it the first time he or she logged in.
- Add privileges to a user's account, because a user needs to perform privileged tasks.
- Change the default login directory for a user's account, because you want to move a user's files to another disk.
- Change the name of an account, because a user's name changes.

Table 2 lists some of the most commonly used Authorize utility commands. Example 3 shows the use of the Authorize utility to display and modify several UAF records.

Table 2 Commonly Used Authorize Utility Commands

Use	Command Format and Example
Getting UAF record information:	
Getting information on an AUTHORIZE command	HELP topic `UAF> HELP ADD`
Displaying full information about all users	SHOW user-name `UAF> SHOW HARKINS`
Writing brief information about all users to the file SYSUAF.LIS	LIST user-name `UAF> LIST HARKINS`
Modifying a UAF record:	
Changing the user name (You must also change the password)	RENAME old-user-name new-user-name/qualifiers `UAF> RENAME JONES SMITH /PASSWORD=NEWPASSWORD`
Changing fields other than the user name	MODIFY user-name /qualifiers `UAF> MODIFY SMITH /PRIVILEGE=VOLPRO`

```
(1) $ SET DEFAULT SYS$SYSTEM
(2) $ RUN AUTHORIZE
(3) UAF> SHOW SMITH
    Username: SMITH                 Owner:  SUE SMITH
    Account:                        UIC:    [200,010]
([FINANCE,SMITH])
    CLI:      DCL                   Tables: DCLTABLES
    Default:  $DISK1:[SMITH]
    LGICMD:   LOGIN
    Login flags:
    Primary days:    Mon Tue Wed Thu Fri
    Secondary days:                      Sat Sun
    No access restrictions
    Expiration:        (none)    Pwdminimum:  0   Login Fails:     1
    Pwdlifetime:       (none)    Pwdchange:               (none)
    Last Login:        (none) (interactive), (none) (non-interactive)
    Maxjobs:            0  Fillm:         20  Bytlm:          4096
    Maxacctjobs:        0  Shrfillm:       0  Pbytlm:            0
    Maxdetach:          0  BIOlm:         18  JTquota:        1024
    Prclm:              2  DIOlm:         18  WSdef:           300
    Prio:               4  ASTlm:         24  WSquo:           512
    Queprio:            0  TQElm:         10  WSextent:       1024
    CPU:          (none)   Enqlm:         30  Pgflquo:       10000
    Authorized Privileges:
      TMPMBX NETMBX
    Default Privileges:
      TMPMBX NETMBX

(4) UAF> MODIFY SMITH /PRIVILEGE=SETPRV
    %UAF-I-MDFYMSG, user record(s) updated
(5) UAF> MODIFY BACH /DEVICE=$DISK2
    %UAF-I-MDFYMSG, user record(s) updated
(6) UAF> MODIFY MATTHEWS /PASSWORD=REMEMBER /PWDEXPIRED
    %UAF-I-MDFYMSG, user record(s) updated
(7) UAF> RENAME TAYLOR WARNER -
    _UAF> /PASSWORD=CLEOPATRA /PWDEXPIRED /DIRECTORY=[WARNER]
    %UAF-I-MDFYMSG, user record(s) updated
(8) UAF> LIST /BRIEF *
    %UAF-I-LSTMSG1, writing listing file
    %UAF-I-LSTMSG2, listing file SYSUAF.LIS complete
(9) UAF> EXIT
    %UAF-I-DONEMSG, system authorization file modified
    %UAF-I-RDBNOMODS, no modifications made to rights database
(10) $ BACKUP $DISK1:[BACH...] $DISK2:[BACH...]
(11) $ RENAME $DISK1:[000000]TAYLOR.DIR WARNER.DIR
    $
```

Example 3 Displaying and Modifying User Accounts

Notes on Example 3.

1. The Authorize utility always looks for SYSUAF.DAT in your current default directory. To affect the system-wide UAF file, SYS$SYSTEM:SYSUAF.DAT, set your default to SYS$SYSTEM.

2. If the SYSUAF.DAT file does not exist, the Authorize utility asks whether you want to create a SYSUAF.DAT file. If you answer YES, it creates one.

   If you know SYS$SYSTEM:SYSUAF.DAT really does exist, answer NO to return to DCL. Then make sure your default directory is SYS$SYSTEM.

3. This SHOW command displays the values in the UAF record for the account SMITH. (By entering the HELP command, you can generate a list of all the Authorize utility commands.)

4. This command adds the SETPRV privilege to the SMITH account.

5. This command changes the login device for the BACH account to $DISK2. The name of the login directory remains the same. (After exiting from the Authorize utility, be sure to move the user's files to the $DISK2 device, see note 10).

6. When you change the password for a user, you should also mark the new password as expired. This requires the user to reset the password when he or she logs in. If the password is not changed, the user will be prevented from logging in again.

7. If a user's name changes, you may be asked to alter the user name of the account. This RENAME command changes the account name, login directory, and password for the TAYLOR account.

   When you change the name of an account, you must also change the password for the account. By including the /PWDEXPIRED qualifier with the RENAME command, you mark the new password as expired. If you change the default login directory in the UAF, remember to rename the directory on the disk (see step 11).

8. This LIST command outputs brief information on each UAF record to the SYSUAF.LIS file. (You can obtain the same listing by typing LIST and pressing the RETURN key.)

9. To exit from the Authorize utility, enter the EXIT command or press CTRL/Z.

10. Changing the login device in a UAF record does not create a directory on the device (see note 5). This BACKUP command copies all of the files in [BACH...] from $DISK1 to $DISK2, preserving their directory structure. Once BACH has logged in to the account to make sure the files on $DISK2 are accessible, you should delete the $DISK1:[BACH...] directory structure.

11. When you ran the Authorize utility, you changed the account name, login directory, and password for the TAYLOR account (see note 7). Now, you must change the name of the directory on the $DISK1 device to match the UAF record. This RENAME command changes the name of the TAYLOR directory to WARNER. (The RENAME command can change a file specification as long as the file remains on the same disk.)

2.3 Deleting a User Account

When a user no longer needs access to the system, you should remove the user's account. Removing an account is a three-step operation. You must:

1. Delete the user's files, or move them to another directory.
2. Delete the user's directory.
3. Remove the user's entry from the user authorization file.

Before starting, you must first decide what to do with the user's files. If you decide to save the files, you may want to move the files to another user's directory or back them up to tape or diskette. If the files are no longer useful, you may simply delete them.

Example 4 shows the steps for removing a user from the system.

```
   $ SET PROCESS /PRIVILEGE=SYSPRV
(1) $ RENAME $DISK1:[JANE]*.*;* $DISK1:[DICK.JANESTUFF]
   $ SET FILE/PROTECTION=S:D $DISK1:[000000]JANE.DIR;1
(2) $ DELETE $DISK1:[000000]JANE.DIR;1
   $ SET DEFAULT SYS$SYSTEM
   $ RUN AUTHORIZE
(3) UAF> REMOVE JANE
   %UAF-I-REMMSG, record removed from SYSUAF.DAT
   %UAF-I-RDBREMMSGU, identifier JANE value: [000200,000002] removed from
   RIGHTSLIST.DAT
   UAF> EXIT
   $
```

Example 4 Removing a User Account

Notes on Example 4.

(1) This RENAME command recatalogs all the files in the JANE directory into a subdirectory of the DICK directory. To gain access to the files in the JANE directory, the SYSPRV privilege may be necessary.

(2) By default, directory files do not permit delete access by any category of user. To remove a directory file, you must change the protection of the directory and you must remove all files from the directory.

This DELETE command removes the JANE directory, since it does not catalog any files and it permits delete access by SYSTEM users.

(3) To remove the UAF record, change your directory to SYS$SYSTEM, invoke the Authorize utility, and enter the REMOVE command. In response to the REMOVE command, the system indicates that it has removed the record from the UAF.

3 The Automatic Login Facility

In addition to creating an account for each user, you can create an account and assign it to a particular terminal for automatic login. Users can log in to the account by simply pressing the RETURN key; no user name or password is required.

To assign an account to a terminal for automatic login, you must:

1. Log in to an account and enable the SYSPRV privilege.
2. Establish an account that does not require a password.
3. Execute the SYS$MANAGER:ALFMAINT command procedure.

The ALFMAINT procedure maintains the automatic login file (SYS$SYSTEM:SYSALF.DAT). This procedure prompts you for a terminal name and a user name. It stores this information in the automatic login file.

Example 5 shows the steps for setting up a terminal for automatic login.

```
❶ $ SET PROCESS /PRIVILEGE=SYSPRV
  $ SET DEFAULT SYS$SYSTEM
  $ RUN AUTHORIZE
❷ UAF> MODIFY WARNER /NOPASSWORD
  %UAF-I-MDFYMSG, user record(s) updated
  UAF> EXIT
❸ $ @ALFMAINT


  Enter the name of the terminal that you would like to set
  for automatic login, or a blank line or EXIT to exit.

  Terminal (ddcu)? TTA1


  Enter the username you would like to automatically
  login on _TTA1:.
  Enter a blank line to display and optionally delete
  the record for _TTA1:.

❹ Username? WARNER

  Terminal _TTA1 user WARNER record added.

  Enter the name of the terminal that you would like to set
  for automatic login, or a blank line or EXIT to exit.

  Terminal (ddcu)? EXIT
  $
```

Example 5 Setting Up a Terminal for Automatic Login

Notes on Example 5.

❶ First, log in to an account and enable the SYSPRV privilege.

❷ Run the Authorize utility to establish an account that does not require a password.

This MODIFY command alters the WARNER account so that no password is needed during login. Since any user will be able to log in to the WARNER account from any terminal without entering a password, the account probably should not be a privileged account.

❸ Finally, to associate the WARNER account with the TTA1 terminal, execute the ALFMAINT procedure. Enter TTA1 for the terminal name.

❹ In this example, WARNER is entered to associate the WARNER account with TTA1.

Now, when any user presses the RETURN key on TTA1, the system will automatically use the information in the WARNER account to create the interactive session.

At a later time, you may want to remove this entry from the automatic login file. If you enter a blank line at this prompt, the system will tell you whether the terminal already has automatic login enabled. If it is already enabled, the system will ask you whether to delete the automatic login entry.

LEARNING ACTIVITY

1. Refer to Chapter 2, **Accounts and Security**, and Appendix **AUTH**, in the *MicroVMS User's Manual* for further information on the AUTHORIZE Utility.

4 Maintaining System-Wide Start-Up Command Procedures

MicroVMS provides a group of start-up command procedures that automatically execute each time you start up the system, and login command procedures that execute each time a user logs in. These procedures configure and initialize the MicroVAX/MicroVMS system to create an efficient working environment.

This section describes some of the features of these start-up procedures. For more information on command procedures, refer to Chapter 7 of the *MicroVMS User's Guide*.

4.1 System-Independent Start-Up Procedure

When the system starts up, the command procedure SYS$SYSTEM:STARTUP.COM is automatically invoked. This is the first of four procedures that configure and initialize the MicroVAX/MicroVMS system.

Operations in this procedure are site-independent; that is, these operations are common to all VAX and MicroVAX systems. Therefore, you should not modify this file. Modifications to this file could jeopardize system updates, to the point of failure.

4.2 System-Specific Start-up Procedure

As system start-up continues, SYS$SYSTEM:STARTUP.COM invokes the command procedure SYS$MANAGER:SYSTARTUP.COM. The operations in this procedure are site-specific; that is, these operations are pertinent to your system alone, and depend on your system's configuration.

The configuration of your MicroVAX/MicroVMS system depends on the hardware and software options you have selected and installed. You must modify the procedure SYS$MANAGER:SYSTARTUP.COM to match this configuration. Modifications can include:

- Defining system-wide logical names.

  These logical names can refer to directory structures, files, command procedures or executable programs. For example, the following command defines the system-wide login command procedure.

```
$
$! Define system-wide login command procedure invoked for all users.
$ DEFINE /SYSTEM /EXEC SYS$SYLOGIN SYS$MANAGER:SYLOGIN.COM
```

- Installing known images by running SYS$SYSTEM:INSTALL.

  You should not confuse the INSTALL utility with the command procedure SYS$UPDATE:VMSINSTAL. VMSINSTAL copies the files from an external source, such as a tape or diskette, onto your system disk. The INSTALL utility is a software tool that enhances the performance of a frequently used program: it speeds up activation of the program, and/or allows a nonprivileged user to execute a program that requires enhanced privileges.

- Starting up layered products.

  Some products require that initial actions be taken each time the system starts up. These actions are dependent on the particular software product. For more information, refer to the installation guide that accompanies each software product.

- Mounting additional fixed disks.

  When the system starts up, it automatically mounts the system disk. If the system configuration includes additional fixed disks, they must be mounted at system start-up to make them available to all users. The MOUNT command invokes the MOUNT utility, which ensures that disks are not already allocated, and are physically present. For more information on the MOUNT command, refer to Appendix **DCL** in the *MicroVMS User's Manual*.

- Initializing batch and print queues. This topic is discussed later in the module.

4.3 Modifying SYS$MANAGER:SYSTARTUP.COM

You can modify SYS$MANAGER:SYSTARTUP.COM in two ways:

1. By removing comment delimiters from commands already contained in the procedure.
2. By adding new commands to the procedure.

Example 6 describes both types of modification.

```
        .
        .
        $!
(1)     $  RUN SYS$SYSTEM:INSTALL
(2).    sys$system:mail                 /open/header/priv=(sysprv,oper,world,netmbx)
        sys$system:phone                /priv=(netmbx,oper,prmmbx,world,sysnam)
        sys$system:rtpad                /priv=(tmpmbx)
        $!sys$system:analimdmp          /priv=(cmexec,cmkrnl)
        $!sys$system:monitor            /priv=(cmkrnl)
        $!sys$system:request            /priv=(tmpmbx)
        $!sys$system:submit             /open/header/priv=(tmpmbx)
        $!sys$system:tpu                /open/header
        $!sys$share:tpushr              /open/header
        $!sys$message:tpumsg            /open/header
        $!
        .
        .
        $!
(3)     $  MOUNT/SYSTEM DISK$WORK3:  VOL_3  WORK3
        $!
        .
        .
```

Example 6 Modifying SYS$MANAGER:SYSTARTUP.COM

Notes on Example 6.

❶ By removing the comment delimiter from this command line, you ensure that the INSTALL utility runs every time the system starts up.

❷ Remove the **$!** comment delimiters to specify those utilities you want installed on the system. Because the lines following the RUN command contain data for the INSTALL utility rather than DCL commands, you must remove the DCL dollar sign prompt as well as the exclamation point.

In this example, the comment delimiters are removed from utilities contained in the Common Utilities Option (MAIL, PHONE, and Remote Terminal Support).

You may also have to change the order of the lines so that the lines without comment delimiters appear immediately after the INSTALL command line, and before the remaining comment lines. You can compare this example with the default SYS$MANAGER:SYSTARTUP.COM supplied with your system.

❸ This MOUNT command, added to SYS$MANAGER:SYSTARTUP.COM, makes DISK$WORK3 available to all users on the system, ensures that the proper disk is present, and assigns it the logical name WORK3.

5 Maintaining a System-Wide Login Command Procedure

In addition to the system start-up command procedures, MicroVMS provides a system-wide login command procedure. The procedure, SYS$MANAGER:SYLOGIN.COM, is automatically invoked whenever a user logs on to the system. It executes before the user's personal LOGIN.COM is invoked. Like LOGIN.COM, it can be used to set up DCL symbols and other characteristics of the DCL environment.

Like the system start-up procedures, SYS$MANAGER:SYLOGIN.COM performs actions that affect all users on the system. You can modify this file to tailor the working environment and to simplify a user's interaction with the system.

5.1 Modifying SYS$MANAGER:SYLOGIN.COM

Example 7 shows the command procedure SYS$MANAGER:SYLOGIN.COM provided with the MicroVMS software. The accompanying notes describe the procedure's actions and indicate where and how you can make modifications to the file.

```
$! Enable control-T for interactive processes.
$
$  IF F$MODE() .EQS. "INTERACTIVE" THEN SET CONTROL=T
$
$! For the system manager's account, define some special symbols.
$
(1) $   ACNT_NAME = F$GETJPI(0,"USERNAME")
$  ACNT_NAME = F$EDIT(ACNT_NAME,"COLLAPSE")
$  IF ACNT_NAME .NES. "SYSTEM" THEN GOTO ALL_ACNTS
$
(2) $  ! Define shutdown commands
$
$  SHUTDOWN  == "@SYS$SYSTEM:SHUTDOWN 0 SHUTDOWN YES NO LATER NO NONE"
$  REBOOT == "@SYS$SYSTEM:SHUTDOWN 0 SHUTDOWN YES NO LATER YES NONE"
$  SHUTDOWN1 == "@SYS$SYSTEM:SHUTDOWN"
$
$  ! Define command for removing a MicroVMS option from the
$  ! system disk.
$
$  REMOVE == "@SYS$UPDATE:REMOVE SYS$UPDATE:VMSKITBLD.DAT"
$
$
(3) $ALL_ACNTS:
$
$  INSTALL :== $INSTALL/COMMAND
$  MOU*NT :== MOUNT/NOASSIST
$  IF F$MODE() .EQS. "INTERACTIVE" THEN SET TERMINAL/INQUIRE
(4) $
$
$DONE:
$  EXIT
```

Example 7 SYS$MANAGER:SYLOGIN.COM

Notes on Example 7.

❶ This portion of the command procedure uses DCL lexical functions to determine whether the user is logged in to the SYSTEM account. Lexical functions are used to get information from within a command procedure. For more on the use of lexical functions, refer to Chapter 7 and Appendix LEX in the *MicroVMS User's Manual*.

If the user is not logged in to the SYSTEM account, execution of the command procedure skips to the line beginning ALL_ACNTS (step 3 below).

❷ This section of the procedure makes symbol assignments. These symbol assignments establish synonyms for DCL commands and command strings. Because they are located in this portion of the command procedure, these assignments apply only to the SYSTEM account. Therefore, you could add commands to this portion of the procedure to further enhance the system manager's working environment.

❸ This portion of the procedure makes symbol assignments that apply to any user logging in to the system. If you want to define additional symbols for all users, place the symbol assignments here.

5.2 SYS$LOGIN:LOGIN.COM

After executing the system-wide login procedure, the system searches for the user's personal login command procedure. This procedure, called LOGIN.COM, is normally located in the user's login default directory. Each user can use this procedure to personalize his or her working environment.

As with the other start-up and login procedures, MicroVMS provides a template file that you or another user can modify. This template, LOGIN.COM, is located in the default directory of the USER account ($DISK1:[USER]).

6 Alternate System Access Methods

Most of the time, you can get access to the MicroVAX/MicroVMS system without trouble. There are situations, however, that may prevent you from logging in to the system. These situations might include:

- All passwords on the system being forgotten.
- All terminals set for automatic logins to nonprivileged or unusable accounts.
- Privileges removed from all accounts.

While it is unlikely that these situations will occur, this section describes the steps you can take to get access to a system when normal logins are prevented.

6.1 Emergency Logins

The preferred method for entering an inaccessible system is through the use of the alternate User Authorization File (UAF).

To log in to the system when normal login is impossible:

1. Press and release the HALT button on the control panel.
2. When the console prompt (>>>) appears on the operator's terminal, type:

   ```
   >>>B/1 DUA0
   ```

   This response initiates a conversational bootstrap, which allows you to alter system parameters before the system starts up.
3. At the SYSBOOT prompt, set the alternate UAF parameter by typing the following:

   ```
   SYSBOOT> SET UAFALTERNATE 1
   SYSBOOT> EXIT
   ```

   If you set the alternate UAF parameter, and no alternate user authorization file exists, the system behaves like an open system; that is, you can get access to the system from the console terminal using any name and password, and any account you log in to has complete privileges.
4. Wait for the system start-up process to complete. This process is described in Example 2 of Module 2, System Start-up and Product Installation. As system start-up proceeds, several informational messages are displayed on the terminal, the final one being:

   ```
   SYSTEM job terminated at 12-SEP-1985 13:44:57.75
   ```

   When this message appears, press RETURN to display the USERNAME prompt.

5. Log in at the console terminal using any name and any password. Your account has all privileges.

6. Fix the problem that prevented normal logins. This may include resetting passwords, restoring user privileges, or resetting automatic logins.

7. After fixing the problem, reset the UAFALTERNATE parameter as follows:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> SET UAFALTERNATE 0
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
```

   These commands ensure that the default user authorization file is invoked the next time the system starts up.

8. Finally, shut down and restart the system by invoking the command procedure SYS$SYSTEM:SHUTDOWN and responding to its prompts.

6.2 Bypassing Start-Up and Login Procedures

The system start-up and login procedures provided with the MicroVMS software should always work. If these files are modified, however, error conditions may prevent the system from starting properly.

To start the system without invoking these procedures:

1. Press and release the HALT button on the control panel.
2. At the console prompt, initiate a conversational bootstrap by typing:

   ```
   >>> B/1 DUA0
   ```

3. At the SYSBOOT prompt, type the following:

   ```
   SYSBOOT> SET/STARTUP OPA0:
   ```

   This command specifies that start-up commands will come from the console terminal rather than the system start-up command procedure.

4. Fix the problem preventing logins. This may include editing start-up procedures or modifying accounts. Because the system is not completely initialized, use the following DCL commands to edit the system start-up command procedures.

   ```
   $ ON ERROR THEN CONTINUE
   $ SET DEFAULT DUA0:[SYS0.SYSEXE]
   $ SET TERMINAL/INQUIRE
   $ EDIT DUA0:[SYS0.SYSEXE]SYSTARTUP.COM
   ```

   Notice that, because no logical names are defined, these commands specify the physical device names.

5. Reset the start-up procedure as follows:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> SET/STARTUP SYS$SYSTEM:STARTUP.COM
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
```

   These commands ensure that the corrected start-up procedures execute the next time the system starts up.

6. Shut down and restart the system by invoking the command procedure SYSSSYSTEM:SHUTDOWN and responding to its prompts.

   NOTE

   These procedures allow anyone with physical access to the MicroVAX console terminal to log in as a privileged user. If you require protection of data on your system, make sure you provide adequate physical protection (such as keeping the MicroVAX and its console terminal in a locked or guarded office).

7 Setting Up and Maintaining Print and Batch Queues

As system manager, you are responsible for creating and controlling queues on the MicroVAX/MicroVMS system. Queues manage the system workload, allowing equitable access to system resources. The types of queues are the print queue and the batch queue.

The print queue allows users on the system to produce printed copies of files. If multiple users request files to be printed. the print queue stores the print requests until the printer becomes available.

The batch queue allows users to automatically pass DCL commands to the operating system. These commands may execute at a later time, or run without user intervention.

The start-up procedure SYS$MANAGER:SYSTARTUP.COM contains the commands you need to set up a print queue and a batch queue. To set up the queues, you must remove the comment delimiters that precede the commands. In addition, you may need to modify some of the commands to fit specific characteristics, such as device name or device type, line speed, and page size.

Some systems require more than the basic queues described in SYSTARTUP.COM. For example, if your system configuration contains multiple printers, it needs multiple print queues. Multiple batch queues are also possible.

You can add commands to SYSTARTUP.COM to set up these additional print and batch queues. The commands you add differ from those already contained in SYSTARTUP.COM only in specific details such as device type, logical name, and so on.

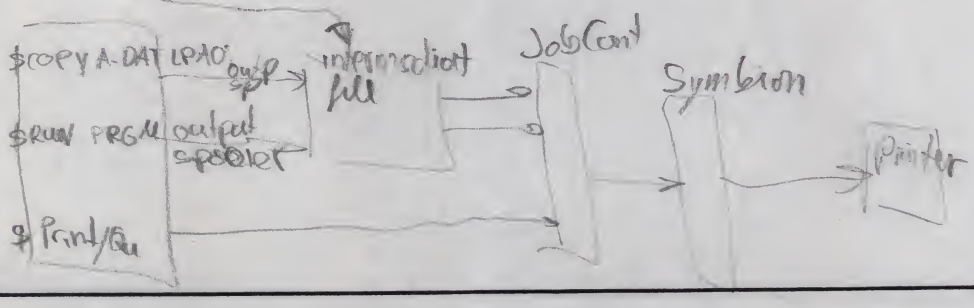
For more information on queues and the commands that affect them, you can refer to the online HELP library. Under the topic "Hints" you can find a category devoted to batch and print jobs.

7.1 Setting Up Print and Batch Queues

Example 8 shows a part of SYS$MANAGER SYSTARTUP.COM, and describes the commands used to set up the print and batch queues on the MicroVMS system. In this example, assume that the print device is a hard-copy terminal connected to terminal port 0.

```
$
$! Remove the comment delimiter in the next command, to start the
$! queue manager, if you are going to have either print or batch queues.
(1) $  START/QUEUE/MANAGER/BUFFER_COUNT=10/EXTEND_QUANTITY=25
$
$! Set up print queue -- commented out below.
$! Remove comment delimiters if you have installed queues from the
$! Secure User Environment Option and want a print queue set up.
$
$! Your print device can be either a line printer or a hard-copy terminal
$! There are different commands depending upon what you use for a print
$! device.
$
$! First define $PRINTER to name the device you will use for printing.
(2) $  DEFINE /SYSTEM $PRINTER  $TERMINAL0    ! if terminal
$! DEFINE /SYSTEM $PRINTER  _LPA0:       ! if line printer
$
$! Set the printer characteristics if you will be printing on a line
$! printer.
$! SET PRINTER /NOWRAP /NOTRUNCATE /CR $PRINTER
$
$! Set the terminal characteristics if you will be printing on a terminal.
$! Some terminal characteristics vary, depending upon the type of terminal
$! being used.  These include:  speed, width, length, and device_type.
(3) $  SET TERMINAL $PRINTER /PERMANENT /NOBROADCAST /NOTYPEAHEAD /NOWRAP -
$  /SPEED=(1200) /WIDTH=(132) /PAGE=(66) /DEVICE_TYPE=(LA100)
$
$
$! Set your printing device spooled to the queue SYS$PRINT.
(4) $  SET DEVICE /SPOOLED=(SYS$PRINT, SYS$SYSDEVICE:) $PRINTER
$
$! Start the queue manager and the print queue.  You may want to specify
$! a flag page if you have a line printer for a printing device.
(5) $  INITIALIZE /QUEUE /START /ON=$PRINTER SYS$PRINT  ! if terminal
$! INITIALIZE /QUEUE /START /ON=$PRINTER SYS$PRINT /DEFAULT=FLAG
$! if line printer
$
$! Set up batch queue -- commented out initially.
$! Remove comment delimiters if you have installed queues from the
$! Secure User Environment Option and want a batch queue set up.
(6) $  INITIALIZE /QUEUE /START SYS$BATCH /BATCH /JOB:3 /BASE_PRIORITY=3
$
```

Example 8 Modifying SYS$MANAGER:SYSTARTUP.COM



Notes on Example 8

1. This command starts the system job queue manager. The queue manager is the part of the operating system that schedules print and batch jobs. This command must be executed before any other queue command. For more information on the START/QUEUE/MANAGER command, refer to Appendix **DCL** of the *MicroVMS User's Manual*.

2. This command assigns the system-wide logical name **$PRINTER** to the device $TERMINAL0. This assignment allows users to refer to the printer without having to know the actual device name. The device name you specify must match the name of the device on your system.

3. This command sets the characteristics for the terminal you have chosen as your printer. Some of the qualifiers listed here, such as line speed or page width, may vary from terminal to terminal. In this example, the qualifier values indicate a line speed of 1200 bits per second, a page width of 132 characters, a page length of 66 lines, and the device type LA100. To determine the correct values for these qualifiers, refer to the user's manual for your terminal.

4. This command sets up **SYS$SYSDEVICE** (the system disk) as an intermediate storage area for text sent to the print queue **SYS$PRINT**. All printers served by queues should be set up in this way. For more on this command, refer to Appendix **DCL** of the *MicroVMS User's Manual*.

5. This command creates and starts up the print queue. It uses the logical name **$PRINTER** to specify the device on which the queue is located, and assigns the name **SYS$PRINT** to the queue.

6. This command creates and starts up the batch queue, and assigns it the name SYS$BATCH. It also assigns some qualifier values to the queue.

For more information on initializing queues, refer to the description of the INIT command in Appendix **DCL** of the *MicroVMS User's Manual*.

7.2 Using the Print and Batch Queues

7.2.1 Printing Files

Once you have set up a print queue, any user can send a text file to a printer with the PRINT command. When you issue the PRINT command, the file you specify is placed in the print queue then sent to the printer as soon as the printer is available. If you submit more than one print job, those jobs waiting to be printed will be placed in the print queue.

The queue manager determines the order in which jobs are printed. Two criteria are used: the job size, and the time of submission. In general, small jobs print before larger jobs, and jobs submitted earlier print first. To obtain a hard copy of a file, use the PRINT command as follows:

```
$ PRINT weekly.txt
Job WEEKLY (queue SYS$PRINT, entry 830) starting on TTA0:
```

For more information on the PRINT command, refer to Appendix **DCL** of the *MicroVMS User's Manual.*

7.2.2 Submitting Batch Jobs

Once you have set up a batch queue, any user can use the SUBMIT command. This command automatically passes commands to the operating system by placing a command procedure in the batch queue. When you submit a batch job, you can delay execution of the procedure until a later time, or allow its execution without your direct interaction. To place a command procedure in the batch queue, issue the SUBMIT command as follows:

```
$ SUBMIT WEEKLY.COM
Job WEEKLY (queue SYS$BATCH, entry 831) started on SYS$BATCH
```

For more information on the SUBMIT command, refer to Appendix **DCL** of the *MicroVMS User's Manual.*

7.2.3 Getting Information About Queues

You can use the SHOW QUEUE command to get information on the system's batch and print queues. Example 9 shows the use of this command.

```
$ SHOW QUEUE/ALL

Printer queue SYS$PRINT, on TTA0:

   Jobname          Username      Entry  Blocks  Status
   -------          --------      -----  ------  ------
(1) no privilege                   826     267  Printing at block 100
   REPORT           HOLMES         833      19  Pending

Batch queue SYS$BATCH

   Jobname          Username      Entry  Status
   -------          --------      -----  ------
(2) MORIARTY         HOLMES         832  Executing
   no privilege                     657  Holding until 11-SEP-1985 16:50
(3) no privilege                    726  Holding until 12-SEP-1985 09:55
   no privilege                     100  Holding until 13-SEP-1985 08:30
```

Example 9 SHOW QUEUE/ALL Output

Notes on Example 9.

(1) These lines indicate that another user's job is currently printing, and user HOLMES' job REPORT is waiting on the print queue.

(2) This line indicates that user HOLMES' batch job MORIARTY is currently executing.

(3) These lines display other jobs waiting on the batch queue for execution at a later time. To display the job name and user name of jobs that do not belong to your UIC group, your account must have either OPER or SYSPRV privilege.

8 Getting Information About the System

As system manager, you need to verify that the system is running correctly. You can also attempt to pinpoint a problem when something goes wrong. MicroVMS provides many methods for examining system activities. Table 3 lists some commands you can use for this purpose. If you are in the habit of using these commands while your system is running well, you are more likely to recognize a problem when one occurs.

Table 3 Monitoring System Activities

Information Displayed	Command
General System Information	
Overview of all processes on the system	$ SHOW SYSTEM
Overview of print queues	$ SHOW QUEUE/DEVICE/ALL
Overview of batch queues	$ SHOW QUEUE/BATCH/ALL
Overview of all devices on the system	$ SHOW DEVICES
Specific Information (Process or Device)	
Interactive users, terminal names, and PIDs	$ SHOW USERS
Information about a specific device	$ SHOW DEVICE/FULL device-code

8.1 Getting Information About System Status

A **process** is the basic entity that performs work on a MicroVAX system. The SHOW SYSTEM command lists all processes currently on the system. The command displays one line of information for each process. Some of the processes run programs for the system and others run programs for users.

The following processes should be present:

- NULL – Executes while no one is using the CPU
- SWAPPER – Helps to manage the main memory
- JOB_CONTROL – Creates sessions for interactive users
- One process for each interactive user

Other processes that may be present:

- Processes associated with layered products such as DECnet
- The process that controls the line printer, called the print symbiont
- Batch jobs

Example 10 shows sample output from a SHOW SYSTEM command.

```
$ SHOW SYSTEM

(1) VAX/VMS V4.2   9-JUL-1985 15:25:28.21                Uptime     0 04:03:37
  Pid     Process Name    State  Pri      I/O       CPU         Page flts Ph.Mem
(2) 00000010 NULL           COM      0        0    0 03:57:37.33          0       0
(3) 00000011 SWAPPER        HIB     16        0    0 00:00:02.64          0       0
    00000014 JOB_CONTROL    HIB      8       68    0 00:00:01.31         83     185
(4) 0000004A USER           LEF      4      112    0 00:00:23:44       1011     366
(5) 0000005B SYSTEM         CUR      4       97    0 00:00:07.55        493     302
    00000101 SYMBOINT_0001  HIB      6      952    0 00:00:53:35        622      45
    00000111 BATCH_1481     COM      2      195    0 00:00:05:14       2783     217
$
```

Example 10 Using SHOW SYSTEM

Notes on Example 10.

(1) The first line of the SHOW SYSTEM command displays the version of the MicroVMS operating system, the node name (if DECnet is running), the current date and time, and the time since the last start-up. The time since the last start-up is in the format **d hh:mm:ss**, representing days, hours, minutes, and seconds.

(2) The NULL process is usually the first process listed in the SHOW SYSTEM display. Information for this process includes the Process Identification number (PID), assigned when the process starts up. The process state is also listed. The process state indicates the readiness of the process to use the CPU. The process state **COM**, or Computable, indicates that the NULL process is available to use the processor.

Other information displayed includes process priority, input and output, CPU time, page faults and physical memory used. For more information on the SHOW SYSTEM command, and the information it displays, refer to Section 4.6.2, Processes, of the *MicroVMS User's Manual.*

(3) The SWAPPER process controls scheduling on the system. Its process state **HIB**, or Hibernate, indicates that the process is currently inactive.

(4) The USER process in this example represents an interactive user. The process state **LEF**, or Local Event Flag, indicates that the process is waiting for some system event, probably user input.

(5) The SYSTEM process represents an interactive user logged in to the SYSTEM account. The process state **CUR**, or Current, indicates that the process is executing in the processor. This state indicates the process that issued the SHOW SYSTEM command.

8.2 Getting Information About Users on the System

The DCL command SHOW USERS lists the interactive processes and their terminal names. The form of this command is:

$ SHOW USERS

Example 11 shows the sample output from the SHOW USERS command.

```
❶ $ SHOW USERS
            VAX/VMS Interactive Users
              9-JUL-1985 15:25:36.21
       Total number of interactive users = 2

❷  Username      Process Name      PID      Terminal
   SYSTEM        SYSTEM          0000005B   OPA0:
   USER          USER            0000004A   TTA0:
   $
```

Example 11 Getting Information About System Status

Notes on Example 11.

❶ The SHOW USERS command displays a list of all the interactive users and their terminal names.

❷ The user name, process name, process identification (PID), and terminal name associated with each interactive process is listed. Usually the process name is the same as the user name. The PID is a unique identifier, associated with the process when it is created.

LEARNING ACTIVITY (OPTIONAL)

For more information on processes and the information displayed by SHOW SYSTEM, read Section 4.6.2, Processes, of the *MicroVMS User's Manual.*

8.3 Getting Information About Devices on the System

The DCL command SHOW DEVICE displays a list of devices on the system, and the status of each device. Table 4 lists device types common on a MicroVMS system.

Table 4 Common Device Types

Code	Device Type
OP	Console terminal
TT or TX	Other terminals
DU	Fixed disk or diskette drive
MU	Tape drive
PU	Disk controller
XQ	Ethernet interface (DEQNA)
RT	Remote user logged in via DECnet

To get brief information about all devices on the system, use the SHOW DEVICE command. To get detailed information about one or more devices, use the SHOW DEVICE/FULL command.

Example 12 shows sample output from SHOW DEVICE commands.

```
$ SHOW DEVICE
               ❶                         ❷
Device            Device            Error     Volume         Free  Trans Mnt
 Name             Status            Count      Label        Blocks Count Cnt
DUA0:             Mounted               0  MICROVMS          10940    29   1
DUA1:             Online                0
DUA2:             Online                0

Device            Device            Error
 Name             Status            Count
TTA0:             Online                0
TTA1:             Online                0
TTA2:             Online                0
TTA3:             Online                0
OPA0:             Online                0
RTA0:             Offline               0
RTA1:             Mounted               0

Device            Device            Error
 Name             Status            Count
PUA0:             Online                2
XQA0:             Online                1
XQA1:             Online                0

$ SHOW DEVICE/FULL DUA0:

❸ Disk DUA0:, device type RD52, is online, mounted, file-oriented device,
      shareable, available to cluster, error logging is enabled.

      Error count                  0   Operations completed               2027
❹     Owner process               ""   Owner UIC                         [1,1]
      Owner process ID      00000000   Dev Prot     S:RWED,O:RWED,G:RWED,W:RWED
      Reference count             20   Default buffer size                 512
❺     Total blocks             60480   Sectors per track                    18
      Total cylinders            140   Tracks per cylinder                  24

      Volume label        "MICROVMS"   Relative volume number                0
      Cluster size                 1   Transaction count                    29
❻     Free blocks              10940   Maximum files allowed             15120
      Extend quantity              5   Mount count                           1
      Mount status            System   Cache name             "_DUA0:XQPCACHE"
      Extent cache size           64   Maximum blocks in extent cache     1094
      File ID cache size          64   Blocks currently in extent cache    486
      Quota cache size             0   Maximum buffers in FCP cache         71

   Volume status: subject to mount verification,
                  write-through caching enabled.
```

Example 12 Getting Information About Devices on the System

Notes on Example 12.

1. The SHOW DEVICE command displays a list of all devices on the system, and indicates the status of each device (offline, online, or mounted).
2. A high value in the error count column may indicate a problem with the device.
3. The SHOW DEVICE/FULL command displays complete information about the device DUA0:. The first line indicates the device name, device type, and status information.
4. The Owner UIC indicates who has mounted the device. For fixed disks, the owner is generally the system. The owner may differ for other devices. For example, you can use this value to determine the owner of a mounted tape.
5. On a MicroVAX system, a block is equivalent to 512 bytes. Total blocks indicates the number of blocks the device contains.
6. Free blocks indicates the number of blocks not yet allocated for storage.

9 Stopping a Process

There are times when terminal output may stop entirely, and the terminal does not respond to your commands. This may result from a number of causes, some more easily remedied than others.

Screen output and terminal response stop if the pre-defined key HOLD SCREEN (or NO SCROLL) is pressed. Pressing the key again restores output.

If a program you are running encounters an error condition, such as an infinite loop, terminal response may stop. To stop the program and return to DCL control, enter CTRL/Y or CTRL/C.

If neither step restores terminal response, you may find it necessary to stop the process. Other reasons for stopping a process include:

- Stopping a user's process if he or she neglects to log out.
- Stopping an interactive process on a specific terminal, in order to make that terminal available.
- Stopping a batch job if it enters an infinite loop.
- Stopping processes that are created but not terminated by a faulty application program.

You need special privileges to control processes with a UIC different from that of your process. If you enable GROUP privilege, you can affect any process in the same UIC group. If you enable WORLD privilege, you can affect any process on the system. (To determine the UIC of a process, use the SHOW SYSTEM/FULL command.)

To stop a process, log in to a privileged account. Next, enter the SHOW USERS or SHOW SYSTEM command to determine the PID of the process you want to stop. Then, to stop the process, enter the STOP/IDENTIFICATION command.

Example 13 shows the steps for stopping a process.

```
(1) $ SHOW USERS

              VAX/VMS Interactive Users
                9-JUL-1985 15:25:36.21
         Total number of interactive users = 2

    Username      Process Name       PID      Terminal
    USERP         USERP            0000005B  OPA0:
    HARKINS       HARKINS          0000004A  TTA0:
(2) $ SET PROCESS/PRIVILEGE=WORLD
(3) $ STOP/IDENTIFICATION=4A
    $ SHOW USERS

              VAX/VMS Interactive Users
                9-JUL-1985 15:25:36.21
         Total number of interactive users = 1

    Username      Process Name       PID      Terminal
    USERP         USERP            0000005B  OPA0:
    $
```

Example 13 Stopping a User Process

Notes on Example 13.

(1) The SHOW USERS command lists the process identification number that the system assigned to the user's process during login. SHOW SYSTEM also gives you the PID, and is necessary if you need to find the PID of a batch job or other noninteractive process.

(2) WORLD privilege allows your process to affect any process on the system.

(3) This STOP command deletes the HARKINS process, which has the process identification number 4A. (You do not have to include leading zeros in the value you supply to the /IDENTIFICATION qualifier.)

10 Backing Up the System

One of the most important duties to schedule on your MicroVMS system is making backup copies of system and user files. You should routinely back up files in case the originals are deleted or destroyed.

On some systems, users are responsible for backing up their own files as they are created or modified. On others, a system manager or operator is responsible for all backups. If you are responsible for backing up all user files, you should perform backups on a regular schedule. For example, you can back up all user files on a weekly basis and back up selected files (those modified in the last 24 hours) on a daily basis.

The online Backup utility is used to save and restore some or all files on disk. You can run the Backup utility while users are logged in to the system. However, no one else should be using the files you want to back up.

Example 14 shows the steps for backing up files that belong to other users.

```
❶ $ SET PROCESS /PRIVILEGE=READALL
❷ $ MOUNT /FOREIGN $TAPE1
❸ $ BACKUP /VERIFY /LIST=18JUL1.LIS
_From: $DISK1:[*...] /OWNER_UIC=[100,1]
_To: $TAPE1:18JUL1.SAV
$ BACKUP /VERIFY /LIST=18JUL2.LIS
_From: $DISK1:[*...] /OWNER_UIC=[100,2]
_To: $TAPE1:18JUL2.SAV
❹ $ DISMOUNT $TAPE1
$
```

Example 14 Backing Up Other User Files

Notes on Example 14.

❶ The READALL privilege enables you to read other user files, even those protected against system access.

❷ The MOUNT command makes the volume available to you for processing and unavailable to other users. Recall that you use the /FOREIGN qualifier with the MOUNT command to manipulate files using the Backup utility.

❸ After mounting the tape cartridge, enter one BACKUP command for each user account, specifying the UIC of the account. In this example, two users have accounts on the system. The first BACKUP command copies all files that have a UIC of [100,1] from $DISK1 to the 18JUL1.SAV save set on $TAPE1. The second BACKUP command copies all files that have a UIC of [100,2] from $DISK1 to the 18JUL2.SAV save set on $TAPE1. (You can also enter the UIC value to the /OWNER_UIC qualifier in alphanumeric format. An example of a UIC in a string format is: [FINANCE,JANE].)

The /VERIFY and the /LIST qualifiers were entered with each BACKUP command. The /VERIFY qualifier directs the system to compare the files in the save set to the original files. The /LIST qualifier directs the system to create a text file listing the files processed during the backup session.

❹ The DISMOUNT command makes the tape drive available to other users.

11 Restoring the System Disk

Sometimes a disk becomes damaged, and you may have to use backup volumes to create a new disk. At other times, a user will accidentally delete a file, and you will have to restore it to the user's directory. Both of these situations require you to restore files from backup save sets.

If the system disk fails so that files cannot be saved:

1. Have the disk repaired or replaced.
2. Reinstall MicroVMS from the distribution kit. Install all the options you need.
3. Install any optional products from their distribution kits.
4. Restore all user files as in Example 15.

To restore user files, use the BACKUP command. Be careful about how you enter the output specifier, because it determines where the files will be restored. Table 5 describes the four types of output specifiers used in the BACKUP command.

Table 5 Formats for Output Specifiers

Function	Format for Output Specifier	Example

Copy directory structure and files from a save set to the output device in their original form.	`[*...]`	`$ BACKUP`
		`_From: $TAPE1:MINE.SAV`
		`_To: $DISK1:[*...]`

Copy directory structure and files from a save set to your current directory and subdirectory.	`[...]`	`$ BACKUP`
		`_From: $TAPE1:MINE.SAV`
		`_To: [...]`

Copy directory structure and files from a save set to the named directory, and subdirectories of the named directory.	`[dir-name...]`	`$ BACKUP`
		`_From: $TAPE1:MINE.SAV`
		`_To: [HARKINS...]`

Copy all files from a save set to the named directory.	`[dir-name]`	`$ BACKUP`
		`_From: $TAPE1:MINE.SAV`
		`_To: [HARKINS]`

```
❶ $ MOUNT/FOREIGN $TAPE1
❷ $ BACKUP/VERIFY $TAPE1:18JUL1.SAV $DISK1:[*...]
  $ BACKUP/VERIFY $TAPE1:18JUL2.SAV $DISK1:[*...]
❸ $ DISMOUNT $TAPE1
  $
```

Example 15 Restoring User Files

Notes on Example 15.

❶ The MOUNT command makes the volume available to you for processing and unavailable to other users. You must include the /FOREIGN qualifier with the MOUNT command to manipulate files using the Backup utility.

❷ This BACKUP command duplicates the directory structure recorded in the save set on the output volume, and creates any necessary directories and subdirectories on the output volume. It also restores all files contained in the save set to their corresponding directories on the output volume.

The /VERIFY qualifier directs the system to compare the files on the output volume with the files in the save set.

❸ The DISMOUNT command makes the tape cartridge drive available to other processes.

LEARNING ACTIVITIES

1. Read the sections **System Backups by UIC** and **Complete System Backups** in Chapter 1 of the *MicroVMS User's Manual*. These sections discuss schemes the system manager can use to maintain complete backups.
2. (Optional) Refer to Appendix **DCL** in the *MicroVMS User's Manual* for further information on the Backup utility.

Summary

- To perform tasks that require privilege:
  - Use an account that is authorized to use privileges.
  - Use SET PROCESS /PRIVILEGE to set privileges.
- Each user needs:
  - A record in the user authorization file SYS$SYSTEM:SYSUAF.DAT
  - A default login directory
- You use the Authorize utility to add, modify, or delete user accounts.
- To delete an account:
  - Save the important files that belong to the account
  - Delete the directory structure associated with the account
  - Remove the account from the system using AUTHORIZE
- Use SYS$MANAGER:ALFMAINT.COM to set a terminal for automatic login.
- Reboot the system to modify UAFALTERNATE if users cannot log in.
- Use SHOW SYSTEM to find out the status of the system.
- Use SHOW USERS to find out which users are logged in.
- If necessary, use STOP /ID=process-id to stop a user process.
- If you are responsible for backing up all user files on a regular basis, back them up by UIC.
- If you lose all the files on the fixed disk, you can reinstall the system and restore all user files from backup copies.

SETTING UP AND USING A NETWORK

Introduction

With multiple MicroVAX II systems, you gain the advantages of **distributed processing**: putting computer power everywhere it is needed. You can make your computing environment more flexible by forming a **network**, which is a group of computer systems that can communicate with each other. A network has two basic components:

1. Hardware that provides a physical connection between systems
2. Software that sends messages between systems

DIGITAL supports several methods of connecting systems, with a wide variety of network configurations. This module presents the steps for incorporating MicroVAX II systems into a network using Ethernet hardware. It also refers you to documentation for more information on other hardware configurations.

Whatever hardware you use, your system communicates with other systems on the network using the software called **DECnet**. This module discusses how to install and start up the DECnet software.

The purpose of a network is to allow users of multiple systems to share data and resources. This module discusses important user-level DECnet functions. It covers remote file access, backing up files across the network, and communication with other network users.

Objectives

To properly manage a MicroVAX/MicroVMS system as a DECnet node, you should be able to:

- Plan a network configuration
- Install the DECnet software
- Configure the network database on each system
- Start up and shut down DECnet
- Use MAIL over the network
- Copy and back up files over the network

Resources

1. *MicroVMS User's Manual*
2. *Guide to Networking on VAX/VMS*
3. *Guide to Local Area Networks*
4. *Networks and Communications Buyer's Guide*

1 DECnet Capabilities

Attaching your system to a DECnet network lets it communicate with other systems. Each system connected to a network and running the DECnet software is called a **node**. When you log in to a DECnet node, the system you log in to is called the **local** node. Other systems on the network are called **remote** nodes. As a user, you can take advantage of the following features of DECnet:

- You have access to files on remote nodes, just as if the files were on the local node. For example, you can use the TYPE command to display a remote file at your terminal. Later in this module, there are examples of file access on remote nodes.
- You can use the Mail utility to send mail to users on remote nodes.
- You can communicate with users on remote nodes via the Phone utility.
- If you have an account on a remote node, you can log in to that node from your local system by means of DECnet, instead of using a terminal directly connected to the remote node.
- You can write programs that communicate with programs running on other nodes (not covered in this course).

2 Planning Your Network

2.1 DECnet Hardware

Once you decide to set up a network, you can choose from several types of hardware for connecting your systems.

2.1.1 Ethernet

A **Local Area Network (LAN)** is a group of interconnected systems located within a limited area. **Ethernet** is the local area network that DIGITAL supports. You can connect many systems to a single Ethernet. An Ethernet is typically used over part of a building, an entire building, or a group of buildings. Any system in the VAX family, as well as other DIGITAL systems, can be connected to an Ethernet. Because Ethernet is a commonly used method for connecting MicroVAX systems, the examples in this course deal with an Ethernet.

The heart of an Ethernet network is the Ethernet cable. To communicate by means of Ethernet, each MicroVAX II must have an Ethernet-to-Q-Bus communications controller, called a DEQNA. The DEQNA module is installed inside the MicroVAX cabinet, and has a cable connecting it to the Ethernet cable. Figure 1 shows an Ethernet cable with several MicroVAX II systems connected to it.

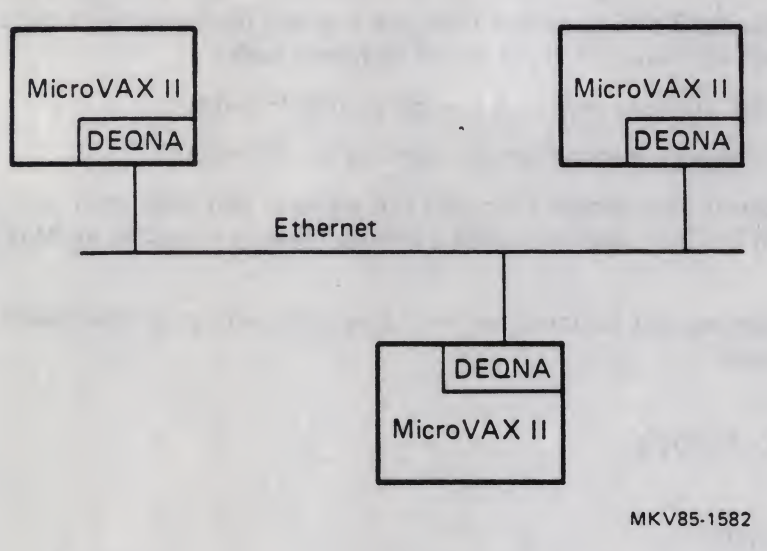


Figure 1 Systems Connected to an Ethernet

2.1.2 Asynchronous Communications

Ethernet is not the only media for DECnet communications. DECnet can also use an asynchronous communications interface, which is the same interface you use to connect user terminals to the MicroVAX. You can connect a cable from a terminal connector on your MicroVAX II directly to a terminal connector of another VAX or MicroVAX. You can also connect a terminal line to a modem; the modem can communicate over a telephone line with another modem connected to another VAX or MicroVAX, giving you DECnet communications over any distance. This course does not explain how to set up the DECnet software for communications over an asynchronous line; see the *MicroVMS User's Manual* for instructions on how to do this.

2.1.3 Synchronous Communications

A third communication media is a synchronous interface. The MicroVAX II uses the DMV11 synchronous interface. It can be connected to a modem for long-distance communication with MicroVAX or other VAX systems.

Your network can combine these types of hardware. For example, you can connect all the systems in your building to a single Ethernet cable; one or more of these systems can also have a synchronous interface connected to a modem, so that a system on the Ethernet can communicate with systems located far away (perhaps on other Ethernets). This course does not tell you how to set up the DECnet software in such a complicated network. You can get more information from the *Guide to Networking on VAX/VMS* or from courses in the DIGITAL data communications curriculum.

2.2 Network Security

Another aspect of network planning is deciding how secure your system should be. On a system that is not part of a network, you and your users are probably aware of who is permitted to log in to the system, and which users have access to which files. However, when a system belongs to a DECnet network, its files can be made accessible to users on remote systems; it may be harder to tell who has access to files on your system.

If the users of your node routinely share programs and data among themselves, you may also want users on other nodes to have easy access to files on your node. However, you may not know who the users of a remote node are, especially if another person manages that node. You and your users can choose to protect files from unknown users on remote nodes.

The paths by which remote users can gain access to files on your node are:

- The default DECnet account
- Explicit access control
- Proxy login

2.2.1 The Default DECnet Account

Files on your node can be made available to remote users through an account called the **default DECnet account**. No one should be able to log in to this account; only the DECnet software can use it. The account should have its own UIC group and no special privileges. If a file on your node is world-readable, a user on a remote node can read it because the default DECnet account has read access to it. If a file is not world-readable, a user on a remote node cannot read it because the default DECnet account cannot read it. Later, this module shows how to cause a default DECnet account to be created.

2.2.2 Access Control Strings

Another way for a remote user to gain access to files is by knowing the user name and password of an account on your node. The UIC and privileges of this account define which files are accessible to the remote user. The remote user includes this user name and password as part of the file specification (called an **access control string**). This module discusses how to specify an access control string.

2.2.3 Proxy Accounts

The third way to give remote users access to your files is by setting up **proxy accounts**. A proxy account is an account on your node that is associated with a specific user or users on a remote node. A remote user does not need to know the password for the proxy account, and gains access to files as if logged in to the proxy account. Appendix **AUTH** of the *MicroVMS User's Manual* documents the commands for setting up proxy accounts with the Authorize utility.

When you set up your network, you choose whether to create a default DECnet account. You can also designate accounts for file access by remote users; they can be accounts belonging to your users, or they can be accounts used only for remote file access. You can either give out the user name and password for such an account, so a remote user can use them in an access control string, or define it as a proxy account so the remote user does not have to specify the user name and password.

In addition to remote file access, DECnet also allows remote users to log in to your node just as if their terminals were physically connected to it. Of course, anyone who attempts to log in to your node must know a valid user name and password, whether logging in by means of DECnet or seated at one of your terminals. Remember to add passwords to the SYSTEM, USER, and USERP accounts; otherwise, any user in the network can log in to these accounts.

If your users need to protect their files, let them know whether remote users have access to files not protected against the world. Users should also prevent their passwords from falling into the hands of anyone with access to the network.

If you use an access control list (ACL) to specify protection for a file or directory, you can use the NETWORK identifier to grant or revoke access to remote users on the network. Individual users can use this method to protect their own files. The *MicroVMS User's Manual* has more information on ACL protection.

NOTE

These guidelines should keep your system safe from casual browsers. Protecting your system against knowledgeable and/or hostile users is more difficult, especially if you do not manage every system in the network. If your site has strict security requirements, you need to understand the guidelines in the *Guide to VAX/VMS System Security*, which require knowledge of concepts in the *Guide to Networking on VAX/VMS*.

2.3 Choosing Names and Addresses

Before you set up a DECnet node, you must assign it a name that is unique within the network. The name can have up to six alphanumeric characters.

You must also give each node a unique address. The address consists of an **area number** from 1 to 63, and a **node number** from 1 to 1023, which is the number of the node within the area. You usually specify the address in the form area.node. For example, in the address 2.101, 2 is the area number and 101 is the node number.

Nodes connected to a single Ethernet are typically assigned to the same area. The examples in this module assume that all your nodes are in the same area. You can choose any area number you like, or use the default area of 1. Larger and more complex DECnet networks can be divided into multiple areas, as explained in the *Guide to Networking on VAX/VMS*.

Within an area, you can choose node numbers arbitrarily as long as they are unique. In the examples in this module, node numbers are consecutive and start at 1. There should be one person who keeps track of the names and addresses of the nodes in the network to make sure no two nodes ever have the same name or address.

The examples in this module use a five-node network with the following nodes:

1. ADMIN1 – A system in the administrative department
2. ADMIN2 – Another system in the administrative department
3. WORDS – A system in the word processing department
4. RAMONA – A single-user system
5. JOHN – A single-user system

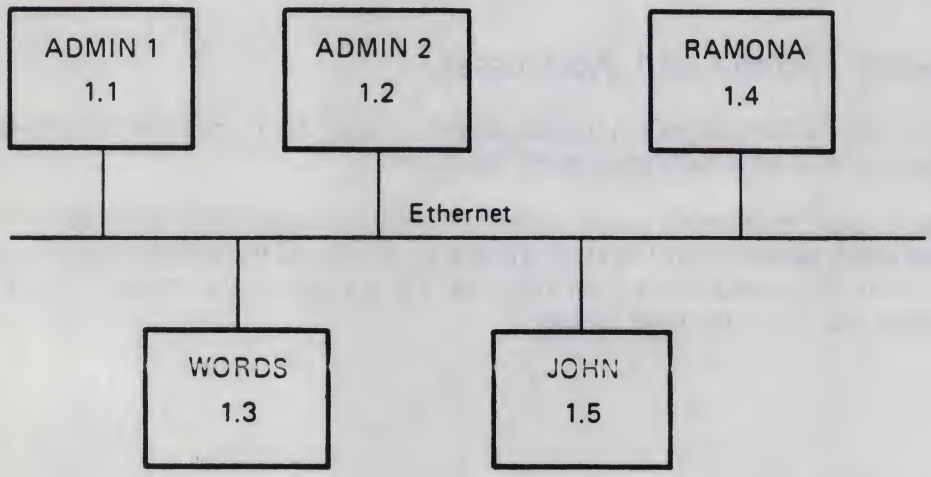


Figure 2 A Five-Node Network

3 Setting Up the Network

3.1 Installing DECnet Software

The DECnet software is packaged as the NET option of the MicroVMS distribution kit. As with any other option, you use VMSINSTAL to install the software. You must also purchase a DECnet license. The license comes with a DECnet **key**, another piece of software you install with VMSINSTAL. The key allows DECnet to run.

Every node in a DECnet network is either:

- A routing node, which can act as an intermediate node when there is no direct path between two given nodes
- A nonrouting node (end node), which can send and receive messages but performs no routing of a message

DECnet running on an end node consumes less CPU time than DECnet running on a routing node, so it is desirable to operate nodes as end nodes. However, some DECnet configurations require certain nodes to be routing nodes. If your network consists of a single Ethernet, you do not need any routing nodes because any node can send messages directly to any other node on the Ethernet.

When you purchase a DECnet license for each node, you purchase either:

- An end-node license, which allows the node to be an end node only, or
- A full function license, which allows the node to be a router or an end node and is more expensive than an end node license

To install the DECnet software, you must follow these steps on every system in the network:

1. Use VMSINSTAL to install the NET option. If you have a TK50 distribution, the option is on the MicroVMS distribution tape. If you have an RX50 distribution, the NET option takes up three diskettes.
2. Use VMSINSTAL to install the DECnet key. The key is on a separate tape cartridge or diskette. The cover letter accompanying your DECnet license contains more specific information on installing the key.

3.2 Using NETCONFIG.COM

After you install the DECnet software, you must configure the system by giving it information about the rest of the network. Each node stores information about the entire network, such as the names and addresses of the other nodes, in a file called the **permanent database**. SYS$MANAGER:NETCONFIG.COM is a command procedure that automatically places most of the necessary information into the database.

Execute NETCONFIG.COM as follows:

```
$ @SYS$MANAGER:NETCONFIG
```

NETCONFIG.COM prompts you for:

1. The name of your node.
2. The address of your node.
3. Whether to create a default DECnet account. As described in Section 2.2, this account makes all world-accessible files on your node accessible to remote users.
4. Whether to operate as a routing node. Because the examples in this module deal with a single Ethernet, none of the nodes need to be routing nodes.

Example 1 shows the use of NETCONFIG.COM on node JOHN.

```
❶ $ @SYS$MANAGER:NETCONFIG.COM

        DECnet-VAX network configuration procedure

This procedure will help you define the parameters
needed to get DECnet running on this machine.  You
will be shown the changes before they are executed,
in case you wish to perform them manually.

❷ What do you want your DECnet node name to be?          : JOHN
❸ What do you want your DECnet address to be?            : 1.5
❹ Do you want a default DECnet account?             [YES]: YES
❺ Do you want to operate as a router?  [NO (nonrouting)]: NO

    Here are the commands necessary to set up your system.

-------------------------------------------------------------

❻ $ RUN SYS$SYSTEM:NCP
    PURGE EXECUTOR ALL
    PURGE KNOWN LINES ALL
    PURGE KNOWN CIRCUITS ALL
    PURGE KNOWN LOGGING ALL
    PURGE KNOWN OBJECTS ALL
    PURGE MODULE CONFIGURATOR KNOWN CIRCUITS ALL
$ DEFINE/USER SYS$OUTPUT NL:
$ DEFINE/USER SYS$ERROR NL:
$ RUN SYS$SYSTEM:NCP    ! Remove existing entry, if any
    PURGE NODE 1.5 ALL
    PURGE NODE JOHN ALL
$ RUN SYS$SYSTEM:NCP
    DEFINE EXECUTOR ADDRESS 1.5 STATE ON
    DEFINE EXECUTOR NAME JOHN
    DEFINE EXECUTOR MAXIMUM ADDRESS 458
    DEFINE EXECUTOR TYPE NONROUTING IV
    DEFINE EXECUTOR NONPRIVILEGED USER DECNET
$ DEFINE/USER SYSUAF SYS$SYSTEM:SYSUAF.DAT
❼ $ RUN SYS$SYSTEM:AUTHORIZE
    ADD DECNET /OWNER="DECNET DEFAULT" -
        /PASSWORD="" -
        /UIC=[376,376] /ACCOUNT=DECNET -
        /DEVICE=SYS$SYSDEVICE: /DIRECTORY=[DECNET] -
        /PRIVILEGE=(TMPMBX,NETMBX) -
        /FLAGS=(CAPTIVE) /LGICMD=NL: -
        /NOBATCH /NOINTERACTIVE
$ CREATE/DIRECTORY SYS$SYSDEVICE:[DECNET] /OWNER=[376,376]

$ RUN SYS$SYSTEM:NCP
    DEFINE LINE    QNA-0 STATE ON
    DEFINE CIRCUIT QNA-0 STATE ON COST 4
    DEFINE LOGGING MONITOR STATE ON
    DEFINE LOGGING MONITOR EVENTS 0.0-9
    DEFINE LOGGING MONITOR EVENTS 2.0-1
    DEFINE LOGGING MONITOR EVENTS 4.2-13,15-16,18-19
    DEFINE LOGGING MONITOR EVENTS 5.0-18
    DEFINE LOGGING MONITOR EVENTS 128.0-4

-------------------------------------------------------------

❽ Do you want to go ahead and do it? [YES]: YES
```

Example 1 Using NETCONFIG.COM

Notes on Example 1.

1. This command invokes NETCONFIG.COM.
2. Specify the name you have selected, in this case JOHN.
3. Specify the address you have selected, in this case 1.5 (area 1, node 5).
4. If you want a default DECnet account, enter YES. YES is the default, so pressing the RETURN key is sufficient.
5. In the example configuration, none of the nodes need to be a router. NO is the default, so pressing the RETURN key is sufficient.
6. NETCONFIG.COM displays the commands it will execute to set up the DECnet node.
7. Note the characteristics of the default DECnet account: it has its own UIC group, its own directory on the system disk, and minimal privileges.
8. NETCONFIG.COM has not yet executed the commands it displayed. Type YES to execute the commands now.

3.3 Adding Nodes to the Network Database

NETCONFIG.COM cannot perform all the work of configuring your network, because you do not specify the names and addresses of other nodes. You must define each node using the Network Control Program (NCP). To execute NCP, you enter the command:

```
$ RUN SYS$SYSTEM:NCP
```

At the NCP> prompt, enter commands of the form:

```
NCP> DEFINE NODE node-address NAME node-name
```

This defines the node in the permanent database. Example 2 shows how to use NCP on node RAMONA to define the rest of the nodes in the network. Note that the managers of other nodes in the network must also use NCP to define RAMONA in their permanent databases.

```
$ RUN SYS$SYSTEM:NCP
NCP>DEFINE NODE 1.1 NAME ADMIN1
NCP>DEFINE NODE 1.2 NAME ADMIN2
NCP>DEFINE NODE 1.3 NAME WORDS
NCP>DEFINE NODE 1.5 NAME JOHN
NCP>EXIT
$
```

Example 2 Defining Nodes in the Permanent Database

If there is a node in the network that has all of the nodes defined in its database, you can copy all of the node definitions to your system with a single command. The above example leaves RAMONA with definitions of all the nodes in the network. If you then install DECnet on node JOHN, you can issue this command on JOHN:

```
NCP> COPY KNOWN NODES FROM RAMONA TO PERMANENT
```

instead of defining each node with a separate NCP command.

4 Starting Up and Shutting Down DECnet Software

4.1 Starting Up DECnet

Installing the DECnet software on your system does not automatically cause the system to participate in the network. After you install DECnet, you need to start it. If DECnet is not running, your users will see error messages if they try to use remote files, send MAIL to remote users, or perform other network operations. Remote users will see error messages when they attempt network operations that involve your node.

SYS$MANAGER:STARTNET.COM is the command procedure that starts DECnet. After you have configured your network database, enter the following command:

```
$ @SYS$MANAGER:STARTNET
```

STARTNET.COM can take up to several minutes to execute, depending on the size of your network database.

You probably want DECnet to start automatically whenever your system starts up. The file SYS$MANAGER:SYSTARTUP.COM that comes with your system contains the previous command, preceded by a comment character (!). Edit SYSTARTUP.COM to remove the comment character from this statement.

4.2 Shutting Down DECnet

When you shut down your system, the DECnet software on your system shuts down automatically. There are times when you want to shut down the DECnet software on your system without shutting down the entire system. For example, when you use VMSINSTAL to install optional software, you should make sure no one has access to your system disk during the installation. Besides forcing local users to log out, you should shut down DECnet to deny remote users access to your system disk.

Shutting down DECnet removes your system from the network without otherwise affecting the system or the rest of the network.

To shut down DECnet on your system, you use the NCP command SET EXECUTOR STATE SHUT. Example 3 shows this command.

```
$ RUN SYS$SYSTEM:NCP
NCP>SET EXECUTOR STATE SHUT
NCP>EXIT
$
```

Example 3 Shutting Down the Network

5 Network User Operations

Once the network hardware is installed and the DECnet software is running, users can take advantage of DECnet's features. To do so, users need to know the names of the nodes in the network. Users do not need to know what hardware is being used for DECnet communications, or which nodes, if any, are routing nodes. Given the name of a remote node, the DECnet software decides how to direct messages to that node.

5.1 Sending Mail to Other Nodes

With DECnet running, any user can send mail to any other user on the network. At the To: prompt, specify a user name of the form:

node::user-name

Within MAIL, you can use the REPLY command to reply to mail that comes from a remote node. MAIL automatically sends the reply to the node the original message came from.

Example 4 shows how to send mail to a user named MATTHEWS on remote node JOHN.

```
$ MAIL
MAIL> SEND
To: JOHN::MATTHEWS
Subject: Network is up and running
Enter your message below. Press CTRL/Z when complete, or CTRL/C to quit:
I'm sending you this message courtesy of DECnet, to let you know my
MicroVAX II is now in the network! -- Ramona
^Exit^
MAIL>
```

Example 4 Sending Mail to a Remote User

5.2 Remote File Access

You can do most file operations on remote files as if they were local. Recall that a file specification is of the form:

```
device:[directory]name.type;version
```

A complete file specification can include the name of the node on which the file resides:

```
node::device:[directory]name.type;version
```

For example, if you log in to node ADMIN1, you can examine a file on node WORDS (if you have read access to the file) with the following command:

```
$ TYPE WORDS::$DISK1:[REPORTS.FY85]BUDGET.MEM
```

and you can edit that file (if you have write access to the directory) with this command:

```
$ EDIT WORDS::$DISK1:[REPORTS.FY85]BUDGET.MEM
```

If you need frequent access to the file, you can make a copy of it on your local node:

```
$ COPY WORDS::$DISK1:[REPORTS.FY85]BUDGET.MEM
```

You can use logical names defined on the remote node as part of the file name. For example, to examine the site-specific start-up command procedure on node WORDS:

```
$ TYPE WORDS::SYS$MANAGER:SYSTARTUP.COM
```

The default directory for remote file operations is usually the directory belonging to the default DECnet account. NETCONFIG.COM sets up the directory [DECNET] on the system disk as the default DECnet directory. This directory is used if you do not specify a directory for a remote file. For example, you can place a file in the default DECnet directory on node ADMIN1 as follows:

```
$ COPY BUDGET.MEM ADMIN1::
```

Likewise, remote users can copy files to the default DECnet directory on your node. Also, the system creates files named NETSERVER.LOG, which record connections from remote nodes, in this directory. Examine the NETSERVER.LOG files on your system to keep track of network activity, and delete them and other files from the DECnet directory as often as is necessary to free up disk space.

If there is a proxy account set up for you on a remote system, you perform remote file access using commands like the previous ones. You gain access to files as if you were logged in to the proxy account rather than the default DECnet account. The default directory for remote file operations is the same as the default directory for the proxy account, and the system creates NETSERVER.LOG files in that directory.

NOTE

The RENAME command cannot transfer files between nodes. Use the COPY command instead.

5.3 Using an Access Control String

The examples in the previous section work if the remote file is not protected against world access, or if the remote system has a proxy account set up to give you access to the files you need.

To get access to a remote file that is not otherwise accessible to you, you can specify the user name and password of an account on the remote system. You gain access to the file as if you were logged in to that account.

For example, assume user MATTHEWS on node JOHN has the password CATFOOD. If MATTHEWS has his own files protected against world access, you can examine one of those files from another node by specifying an access control string:

```
$ TYPE JOHN"MATTHEWS CATFOOD"::TRAVELFY86.MEM
```

If you must use an access control string, notice that the password is echoed to the screen. You should be careful that other users do not see it. If you use an access control string often for file access on a particular system, the manager of that system should probably set up a proxy account for you instead.

5.4 Backing Up Files Over the Network

If your MicroVAX system is not in a network, you must use floppy diskettes or tape cartridges to store backup copies of files. In a network, you can back up files to a disk that belongs to a remote node. This technique is especially useful if your network contains a large VAX system with plenty of disk space.

In Example 5, assume ADMIN1 is a VAX node with large disks. The example shows how to back up files from WORDS to a save set on ADMIN1.

```
(1) $ BACKUP /VERIFY /LIST=AUG15.LIS  [REPORTS...]
(2) _To: ADMIN1::DUA3:[BACKUPS]WORDS_AUG15.BCK /SAVE_SET
    $
```

Example 5 Backing Up Files Across the Network

alleen naar DISK!

Notes on Example 5.

(1) You can use the same BACKUP command qualifiers and select the same files as when backing up files to a diskette or tape.

(2) You must know the name of the disk on ADMIN1 on which to place the save set. In this case, you specify the physical device name DUA3:. You could also have specified a logical name, which would have to be defined on ADMIN1.

You must also have write access to the directory in which to place the save set, in this case [BACKUPS]. Either the directory could allow write access to the world, or you could have a proxy account on ADMIN1 with write access to the directory.

The qualifier /SAVE_SET is required because the save set is created on a disk.

As with backups to removable media, you can restore selected files or an entire save set from a remote node. Example 6 shows how to restore a single file from the save set created in the previous example.

```
$ BACKUP /SELECT=FY85_BUDGET.MEM
_From: ADMIN1::DUA3:[BACKUPS]WORDS_AUG15.BCK/SAVE_SET
_To: [REPORTS.BUDGET]
$
```

Example 6 Restoring a File Across the Network

The system manager of ADMIN1 should perform regular backups of the disks on ADMIN1. The save sets you create are backed up along with the rest of the files on ADMIN1, giving additional protection to the files from your MicroVAX.

Summary

- A network allows your MicroVAX II to communicate with other systems.
- A network has:
  - Network hardware, such as Ethernet, for connecting systems
  - DECnet software, which allows systems to communicate with each other
- Decide whether remote users can have access to files on your node.
- Choose a name and address for each node in the network.
- Use VMSINSTAL to install:
  - The NET option of MicroVMS, which contains the DECnet software
  - The DECnet key, which allows the DECnet software to run
- Execute SYS$MANAGER:NETCONFIG.COM as the first step in setting up the permanent network database.
- Use the NCP (Network Control Program) command DEFINE to add remote node definitions to the permanent database.
- Use SYS$MANAGER:STARTNET.COM to start DECnet on your node:
  - Manually, when you have finished setting up the database
  - Automatically at start-up, in SYS$MANAGER:SYSTARTUP.COM
- Use the NCP command SET EXECUTOR STATE SHUT to shut down DECnet on your node.
- To perform DECnet user operations:
  - For remote file access, prefix the file name with the node name and an optional access control string (user name and password)
  - To send mail, prefix the user name with the node name
  - To back up and restore files, specify a remote file name as the save set name

LEARNING ACTIVITIES (OPTIONAL)

1. Read the booklet *Introduction to Local Area Networks* for an overview of network topologies, hardware, and protocols.

2. Read these chapters of the *Guide to Networking on VAX/VMS:*

   - **Overview of DECnet-VAX and VAX PSI**
   - **DECnet-VAX Components and Concepts**

   If you do not plan for your system to communicate by means of a packet switching network, you can skip any section that deals with VAX PSI, X.25, or X.29.

3. If you plan to use an asynchronous communication line for DECnet, read Section 1.8.3.1, Local Static Asynchronous DECnet, of the *MicroVMS User's Manual*. Note that this section uses terminology that may be unfamiliar to you unless you have read the above chapters of the *Guide to Networking on VAX/VMS*. Also read Section 4.1.5, Using a Modem to Create a DECnet Link, in the *MicroVMS User's Manual*.

4. Read the chapter **Security for a DECnet Node** of the *Guide to VAX/VMS System Security*. This chapter contains terminology that may be unfamiliar to you unless you have read the above chapters of the *Guide to Networking on VAX/VMS*.